AFIT/GAE/ENY/91D-14

1

INVESTIGATION OF THE EFFECTS OF INCREASED ORDER COMPENSATORS IN MIXED H_2/H_∞ OPTIMIZATION

THESIS

Scott R. Wells, Captain, USAF

AFIT/GAE/ENY/91D-14

AFIT/GAE/ENY/91D-14

INVESTIGATION OF THE EFFECTS OF INCREASED ORDER COMPENSATORS IN MIXED H_2/H_∞ OPTIMIZATION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Aeronautical Engineering

Scott R. Wells, B.S.

Captain, USAF

December 1991

Approved for public release; distribution unlimited

Preface

Due to the highly theoretical nature of the problem, this work does not deal with the practical application of mixed H_2/H_∞ theory at all. I have enjoyed performing the analytical research, and I trust the reader will be able to see through the theory to the possible future applications of this particular discipline. I am convinced that as the mixed H_2/H_∞ theory becomes more mature, it will provide large breakthroughs in the controls community and find its way into all kinds of applications. Since I have been able to contribute even a small part toward this end, this research has been as fulfilling as it has been trying.

We, as engineers, spend all our energies trying to find some manageable model of the phenomena we observe in nature with the hopes of being able to design something useful. The more I study the sciences (and in this research project in particular) the more impressed I am with the awesome complexity of the natural world in which we live. Even some of the most elementary things still remain a mystery to us. Since I have been able to pull back the veil and reveal a tiny piece of new information, I feel that this thesis is more than just a technical work that will sit on some shelf. It is another chapter in the unfolding testimony of the marvelous God who conceived all of this and brought it into being.

There have been several people who have been crucial, without whom this thesis would not have been possible. First is my thesis advisor, Capt Ridgely.

I would like to thank him for his guidance, patience, and technical expertise. This is really his inspiration, and I am glad I was able to share in it. He devoted more time to his thesis students, just to get us up to speed, than any student could possibly ask. Next, I extend my sincere thanks to my committee members, Maj Mracek and Dr Liebst. Maj Mracek's knowledge of the problem, numerics, and the computer code is was what got me going--and then kept me going. His personal interest in my work is greatly appreciated. Dr Liebst sacrificed many hours and provided a fresh new look at the problem at a point when it looked like all avenues had been exhausted.

Finally, I would like to thank the one who sacrificed more for this thesis than anyone, my wife Sherry. It's one thing for a man to have a wife, but it's a whole other thing for him to have a helpmate and friend. I appreciate her strength and love more than I can possibly say. I look forward to being able to make up lost time with her and my special little boy, Ben.

Scott R. Wells

Table of Contents

List of Figures

List of Tables

Notation

$\mathbf{R}$	field of real numbers
$\mathbf{R}^{nxm}$	set of nxm matrices with elements in $\mathbf{R}$
x^T, A^T	vector/matrix transpose
A^*	complex conjugate transpose of A
$A > 0$ (< 0)	A is positive (negative) definite
$A \geq 0$ (≤ 0)	A is positive (negative) semidefinite
$\sqrt{A}$	matrix square root of A
$\lambda_i(A)$	eigenvalues of A
$\sigma_i(A)$	singular values of A
tr(A)	trace of A $= \sum_{i=1}^{n} a_{ii}$
Im(A)	image of A $\equiv \{y \in F^m \mid y = Ax \text{ for some } x \in F^n\}$
RH_2	space of all real-rational, strictly proper, stable transfer matrices (or vector signals)
RH_∞	space of all real-rational, proper, stable transfer matrices
$\lVert \cdot \rVert_2$	vector or matrix norm on L_2
$\lVert \cdot \rVert_\infty$	matrix norm on L_∞
$I[G(s),\gamma]$	entropy (at infinity) of G(s) at γ

$\left[\begin{array}{c\|c} A & B \\ \hline C & D \end{array}\right]$	transfer function matrix notation $\equiv C(sI-A)^{-1}B + D$
$G^*(s)$	complex conjugate transpose of G(s) $\equiv G^T(-s)$
Ric(M)	Riccati operator on Hamiltonian matrix M
inf	infimum
lim	limit
ln	natural logarithm
$a \equiv b$	a identically equal to b, a defined as b
$\{ A \mid B \}$	set of all A such that B
ARE	Agebraic Riccati Equation
DFP	Davidon-Fletcher-Powell
LFT	Linear Fractional Transformation
LQG	Linear Quadratic Gaussian
LQG/LTR	Linear Quadratic Gaussian with Loop Transfer Recovery
MIMO	Multi-Input, Multi-Output
SISO	Single-Input, Single-Output
$\in$	element of
$\exists$	there exists
$\forall$	for all
$\blacksquare$	end of proof
s	Laplace variable

ω	frequency variable
γ	value of the ∞-norm
γ_o	$\inf_{K \text{ adm}} \lVert T_{ed} \rVert_\infty$
γ_2	$\lVert T_{ed} \rVert_\infty$ when $K(s) = K_{2opt}$
γ^*	$\lVert T_{ed} \rVert_\infty$ when $K(s) = K_{mix}$
α	value of the 2-norm
α_o	$\inf_{K \text{ adm}} \lVert T_{zw} \rVert_2$
α^*	$\lVert T_{zw} \rVert_2$ when $K(s) = K_{mix}$
μ	real number $\in [0,1]$
$\inf_{K \text{ adm}}$	infimize over the set of admissible compensators K
J	performance index
J_μ	performance index at a given μ
$\mathscr{L}$	Lagrangian
$\mathscr{L}_\mu$	Lagrangian at a given μ
K_{2opt}	unique K(s) that gives $\lVert T_{zw} \rVert_2 = \alpha_o$
K_{mix}	a K(s) that solves the mixed H_2/H_∞ problem at some γ
n	order of the plant
n_c	order of the compensator
n^*	optimal order of K_{mix}

AFIT/GAE/ENY/91D-14

Abstract

gamma

The problem of minimizing the 2-norm of one transfer function subject to an ∞-norm bound on another transfer function is examined for increased order controllers. In particular, the theoretical results of the full order case are extended to the higher order case, and SISO and MIMO numerical examples are given for increasingly higher order compensators. Some of the key proofs for higher order compensators include: the global minimum 2-norm is unachievable under output feedback for certain levels of γ regardless of compensator order; the solution to the mixed H_2/H_∞ problem lies on the boundary of the ∞-norm constraint for this same range of γ's; and the suboptimal mixed problem converges to the optimal in the limit for higher order controllers. Also, it is shown that the optimal compensator order for the mixed H_2/H_∞ problem is greater than the order of the plant under certain conditions, and a conjecture about the optimal order for the mixed problem is made.

INVESTIGATION OF THE EFFECTS OF INCREASED ORDER COMPENSATORS IN MIXED H_2/H_∞ OPTIMIZATION

I. Introduction

1.1 Background

Optimal control theory provides powerful tools for designing feedback controllers, particularly when the dynamic system being controlled is very complex. Classical methods such as root locus become too cumbersome or completely unusable for high order, multiple-input multiple-output systems, while optimization techniques can offer a mathematical structure that readily handles these complicated systems. In an optimal control synthesis problem, a controller that will minimize some prescribed cost objective (sometimes with additional constraints) is sought. Extreme care must be taken when defining this cost function because any controller, including one that results in unacceptable time responses (or even an unstable closed-loop system), can be shown to be optimal to some cost function. Obviously, the key in optimal control is defining the "right" cost objective. Two performance measures which are currently receiving a great deal of attention in the controls community are the H_2 and H_∞ norms. These norms are defined by

$$\|G(j\omega)\|_2 \equiv \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{tr}\,[G^*(j\omega)\,G(j\omega)]\,d\omega\right]^{1/2}$$

$$\|G(j\omega)\|_\infty \equiv \sup_{\omega \in \mathbb{R}} \bar{\sigma}\,[G(j\omega)]$$

$$(\,\bar{\sigma} \equiv \text{max singular value}\,)$$

[Dai90]

Both of these performance measures are well motivated and have significant merit. By themselves, however, each produce controllers that have potentially undesirable characteristics. This has led to the development of the mixed H_2/H_∞ optimization problem.

While it is beyond the scope of this work to review the vast amount of literature on H_2 and H_∞ theory, some of the key ideas need to be summarized in order to properly motivate the mixed problem. Consider the general feedback control system shown in Figure 1-1. *P* is the generalized linear, time-invariant plant transfer function and contains all the weighting functions required to obtain this general form. The exogenous and controlled input vectors are *w* and *u*, respectively. *z* and *y* are the controlled and measured output vectors, respectively. Denote the closed-loop transfer function from *w* to *z* as T_{zw}.

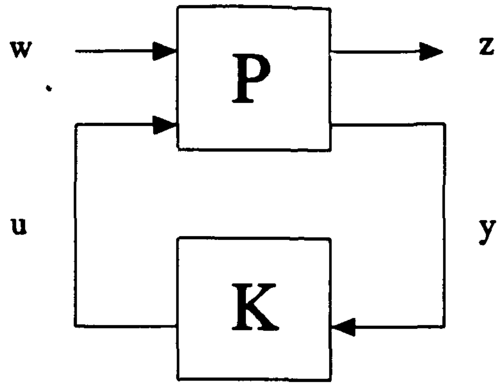

Figure 1-1. H_2 Feedback Control System Block Diagram

If w is assumed to be white Gaussian noise and certain assumptions are made on the plant, the H_2 optimization problem can be formulated and solved. The cost function here is the 2-norm of the closed-loop transfer function T_{zw}. The only compensators that are considered "admissible" in the optimization are those that are real-rational, proper, and internally stabilizing. The problem, stated more formally, is: infimize the 2-norm of T_{zw} over the set of all admissible compensators, or find the K that achieves

$$\inf_{K \text{ adm}} \|T_{zw}\|_2 \equiv \alpha_o$$

Note from the definition of the 2-norm that, given a unit intensity white noise input, (by Parseval's theorem) the square of $\|T_{zw}\|_2$ is equal to the energy of the output signal. The H_2 optimal controller is the one that results in the minimum energy of the controlled output z due to the input w. This is very desirable from a performance point of view for white noise input applications.

It can be shown [Rid91a,206-216] that H_2 optimization, under the assumption of output feedback, is equivalent to a corresponding Linear Quadratic Gaussian (LQG) problem. That is, the compensator that minimizes the LQG cost function

$$J = \int_0^{\infty} (x^T Q x + u^T R u + x^T N u) \, dt$$

is an H_2 optimal compensator. Both problems involve the solution of two Riccati equations and produce a unique controller whose order is equal to the order of the plant (full order). It is well known that while Linear Quadratic Regulators (LQR) and Estimators (LQE) generate systems with guaranteed minimum gain and phase margins, LQG compensators do not exhibit this same quality. In fact, it is possible for an LQG system to have gain and phase margins that are arbitrarily small [Doy78,756-757]. So, while the H_2 optimal controller has desirable performance characteristics, it does not provide any guarantees on system robustness.

It is possible in the LQG problem to recover to some degree the guaranteed margins of the LQR/LQE systems by using a technique called Loop Transfer Recovery (LTR). In LTR, the stability margins are recovered by trading off regulator or estimator performance (depending on where the model uncertainties are entering the system). In LQG/LTR the designer must decide where to break the loop (at the input of the plant or at the output) when defining measures of performance and stability margins. Unfortunately, it is not possible to define performance at one location and stability margins at another [RB86,9.1-9.8]. So, while LQG/LTR is an effective design method for certain cases, it cannot handle the most general case.

Consider now the problem of H_∞ optimization. The generalized block diagram is shown in Figure 1-2.

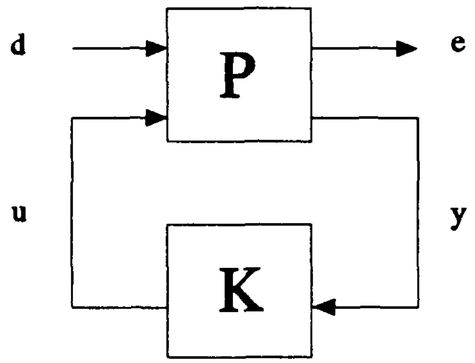

Figure 1-2. H_∞ Feedback Control System Block Diagram

The exogenous input is defined to be the vector *d*, which is assumed to be a bounded energy input. The controlled output vector is denoted by *e*. The problem set-up is exactly the same as before, except now it is the ∞-norm of the closed-loop transfer function T_{ed} that is being minimized; that is find a K which achieves

$$\inf_{K \text{ adm}} \|T_{ed}\|_\infty \equiv \gamma_o$$

The true solution to this optimization problem is usually avoided, and there are several reasons why. First, the H_∞ design algorithms are necessarily iterative, and achieving the exact optimal value is very difficult. Also, H_∞ optimal controllers tend to have some undesirable characteristics. They are typically infinite bandwidth compensators which produce all-pass closed-loop maximum singular value plots [ZDGB90, 2502-2503]. Thus, the true H_∞ optimal solution is not only difficult to calculate, but it may be impractical for a real system. The more practical solution is the H_∞ *sub*optimal controller, which is the compensator that insures

$$\|T_{ed}\|_\infty < \gamma, \text{ where } \gamma > \gamma_o$$

Both the suboptimal and optimal compensators are, in general, non-unique. There are an infinite number of controllers that achieve a given level of H_∞ performance.

One of the key advantages of the ∞-norm is its direct link to system robustness. This is possible because the ∞-norm has the submultiplicative property. That is, for some $F,G \in RL_\infty$, where RL_∞ is the Banach space of all real-rational proper stable transfer matrices,

$$\|FG\|_\infty \le \|F\|_\infty \|G\|_\infty \qquad \text{[Zam66]}$$

Note that the 2-norm does not have this property. To see the application of this property, consider the uncertainty block diagram shown in Figure 1-3. T_{ed} is the nominal closed-loop transfer function, and Δ is a perturbation function that characterizes the system's uncertainty.

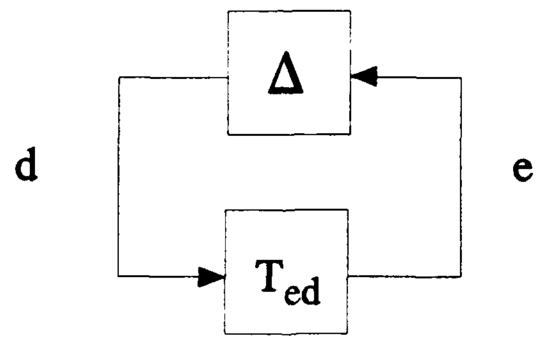

Figure 1-3. Uncertainty Block Diagram

The Small Gain Theorem gives a relationship that describes how large Δ can be before the nominal system becomes unstable.

<u>Small Gain Theorem</u>: Assume $T_{ed}(s), \Delta(s) \in RH_\infty$.

If $\quad \| T_{ed}(s)\ \Delta(s) \|_\infty < 1$

then the closed-loop system (with Δ) is stable [Zam66]

Then, by the submultiplicative property of the ∞-norm,

$$\| T_{ed}(s)\ \Delta(s) \|_\infty \leq \| T_{ed}(s) \|_\infty \ \| \Delta(s) \|_\infty < 1$$

Thus, to ensure system stability,

$$\| \Delta(s) \|_\infty < \frac{1}{\| T_{ed} \|_\infty}$$

The smaller $\| T_{ed} \|_\infty$ is, the more uncertainty the system can handle. Due to the submultiplicative property of the ∞-norm, H_∞ optimization seems like an obvious choice for designing a system for stability robustness.

As can be seen, two reasonable choices for the cost objective in the optimization problem are the 2-norm and ∞-norm of the closed-loop transfer functions. The 2-norm is desirable because it results in optimal (in the energy sense) performance in the face of white noises. The ∞-norm is desirable because it guarantees a level of robustness to plant uncertainties. What would be most desirable would be a methodology that incorporates both. This is what has been termed the mixed H_2/H_∞ optimization problem. Consider now the general form of the feedback control system for the mixed problem as shown in Figure 1-4.

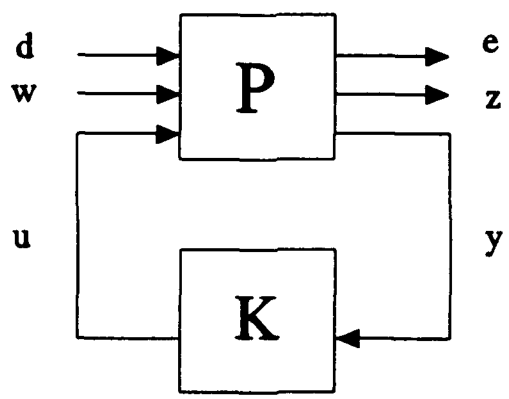

Figure 1-4. Mixed H_2/H_∞ System Block Diagram

The exogenous input d is assumed to be a deterministic signal of unknown but bounded energy. The input w is assumed to be a zero-mean white Gaussian noise of unit intensity. The signals e and z are controlled outputs, and may be equal, dependent, or independent. The signals u and y are the control inputs and measured outputs, respectively. All weighting functions required to obtain this general form are included in the generalized plant transfer function P. Denote the closed-loop transfer functions from w to z and from d to e as T_{zw} and T_{ed}, respectively. Stated formally, the general mixed H_2/H_∞ synthesis problem is to find an admissible controller K(s) that achieves

$$\inf_{K \text{ adm}} \|T_{zw}\|_2, \quad \text{subject to the constraint} \quad \|T_{ed}\|_\infty \leq \gamma$$

As will be shown later, in the region of interest $\|T_{zw}\|_2$ and $\|T_{ed}\|_\infty$ are competing objectives. This makes sense physically since it is expected that some performance should have to be sacrificed in order to gain robustness (and vice versa). The mixed H_2/H_∞ problem provides the needed structure for the designer to explicitly observe and influence this trade-off. An interesting note is that if the problem is set up such that $d = w$ and $e = z$, the closed-loop transfer functions T_{ed} and T_{zw} are equal. The problem is then essentially equivalent to LQG/LTR, and the recovery of the stability margins is seen directly in the trade-off between the 2-norm and ∞-norm. The nice thing about the

H_2/H_∞ approach is that the input and output signals can be defined in any way the designer chooses, so the limitations of LQG/LTR are not present and the problem remains completely general.

The value of designing optimal controllers with mixed H_2 and H_∞ performance objectives has been recognized for some time, but the solution to the most general problem had not been demonstrated until recently by Ridgely in [Rid91a]. Due to the complexity of the problem, his solution requires numerical techniques. In fact, it is generally believed that the problem may not have an explicit analytical solution.

Ridgely was not the first to examine the mixed H_2/H_∞ problem; several earlier papers appeared starting in 1989. The early works by Bernstein & Haddad ([BH89]); Zhou, Doyle, Glover & Bodenheimer ([ZDGB90]); Yeh, Banda & Chang ([YBC90]); Mustafa & Glover ([MG90]); and Khargonekar & Rotea ([KR91]) laid the foundations for the general mixed problem, but made varying assumptions that specialized the more general problem. In addition, they did not minimize the actual 2-norm but rather an upper bound to it. It was not until Rotea & Khargonekar's work ([RK91]) that a special case of the true nonconservative problem was solved. They allowed two sets of inputs and outputs and did not use an upper bound to the 2-norm; however, they did restrict their solution to the case of full state availability for feedback. Finally, Ridgely's work offered the first solutions to the general nonconservative problem. He used a Lagrange multiplier technique and derived the set of necessary conditions for the output feedback case. He also proved some key properties for full order compensators and performed an exhaustive analysis of a SISO and MIMO example, providing valuable insight into the nature of the problem.

While the bulk of Ridgely's work was done with full order controllers, he did take a cursory look at higher order compensators. He found, by example, a higher order controller that satisfied the necessary conditions and produced a 2-norm lower than the corresponding full order compensator. It is obvious from this example and other analysis that the optimal order of the mixed compensator (that is, the lowest order compensator with a minimal realization that achieves the true optimal mixed solution 2-norm) is not the order of the plant. However, the ultimate issue of optimal order was not determined.

1.2 Research Objectives

The work contained here is an extension of Ridgely's dissertation. His solution methodology was used and was applied specifically to compensators whose order is higher than the plant. The main objective of the research was to use the example systems defined in Ridgely's dissertation and numerically solve the mixed H_2/H_∞ for increasingly greater order controllers. An investigation of what these results say about the nature of the problem and what the optimal order might be was then performed. Obviously, these numerical solutions cannot directly prove optimal order in general, but the investigation could shed new light on this difficult subject. In addition to running numerical examples, another objective was to extend some of the analytical proofs for the full order case to the higher order case. Finally, while it was not expected to be able to formally prove the optimal order of the mixed H_2/H_∞ problem, this question motivated all the research, and the proof (or at least a strong conjecture) was always an underlying objective.

1.3 Thesis Outline

This thesis is comprised of six chapters. Chapter I provides the background for this work and outlines the research accomplished. The motivation for doing mixed H_2/H_∞ optimization is presented. Also, a brief history of the problem development is given.

Chapter II gives motivations for investigating higher order compensators and takes an introductory look at the question of optimal order. Some of the problems with determining optimal order are discussed. Then, the optimal orders of related optimization problems are reviewed. Specifically, the optimal orders of compensators in the pure H_2, H_∞, and minimum entropy problems are shown.

Chapter III then begins the formal development of the mixed problem. After defining the problem statement, first-order necessary conditions for the general and suboptimal mixed problems are derived. Then a brief review of the full order analytical results is given.

Chapter IV contains all the analytical proofs that extend the full order results to higher order compensators. The true optimal mixed problem is the main focus of theory presented here; however, the suboptimal mixed problem is also briefly discussed.

Chapter V is the section that contains all the numerical results of the research. First, the algorithm used to numerically solve the set of necessary conditions is discussed. Then, a SISO and MIMO example are given along with discussions of their results.

Finally, Chapter VI concludes the work with a formal conjecture of optimal order supported by both analytical and numerical evidence. Recommendations for future research are offered and a summary is given. The FORTRAN source code of the numerical algorithm discussed in Chapter V is included as an appendix.

II. Higher Order Compensators and Optimal Order

2.1 General Discussion

Why even consider higher order compensators? From a practical point of view, it is unlikely that a higher order controller would ever be implemented on a real system, especially if the plant has any significant size at all. For example, consider a typical aerospace application, aircraft pitch attitude control. The nominal plant from the linearized perturbation equations is fourth order. After adding in all the weights for the input and output signals, the general system plant could easily double. Now, the full order controller must be 8th order. This is already becoming unreasonable from a real-life application point of view. Now, say that the optimal order of a controller for some optimization problem is three times the order of the plant. This would be a 24th order controller. If it is even possible to build such a compensator, it would undoubtedly be prohibitively expensive. It seems the more useful area of interest for practical applications would be reduced order controllers. So again, why consider higher order compensators?

There are at least two very good reasons to examine higher order compensators. The first reason is more philosophical than practical. What is the optimal order of the compensator for the mixed H_2/H_∞ control problem? It has already been shown that it is something greater than the order of the plant, in general. Therefore, if the question of optimal order is to be addressed (which it must be--purely for the sake of fully developing H_2/H_∞ theory), the higher order case must be examined.

The second reason is more practical. The design process is really an art of compromise. The engineer strives to design a system that will meet some set of specifications without over-exceeding the specifications. Why design a system that can handle ten-fold variations in the system parameters when three-fold variations are all that are actually expected? The main reason, besides cost, for not overdesigning a system is that specifications are almost always competing. For example, performance and robustness levels are two specifications that a control system would have to satisfy. Unfortunately, in order to get one, there usually will have to be sacrifices made in the other. In order for an engineer to properly make these trade-offs, he needs to know the limits for the problem, such as the maximum level of performance and the maximum level of robustness that can be achieved by any controller. With these parameters in hand, the engineer can then back off on both ends until a compromise is found that will satisfy both specifications (or determine that the problem has been overspecified). Thus, even if the optimal order of the mixed problem turns out to be infinite (which could never be implemented on a real system), this infinite order compensator represents the limit of achievable performance for a given level of robustness and is important to know. Even more important is the trade-off of performance versus compensator order from full order to the optimal order. This would give the designer the ability to explicitly see what advantages can be gained and how much it will cost to attain them. Therefore, since the optimal order appears to be greater than the order of the plant, increased order compensators must be investigated, even for practical design purposes.

Determining the optimal order of a compensator that satisfies some optimization criterion is not an easy or straightforward task. If the problem can be posed such that compensator order does not have to specified at the beginning of the problem, and the solution ends up defining the compensator order, the optimal order can be determined. However, if a method like Lagrange multipliers is used, the order of the compensator must be chosen during the set-up of the problem. This has the advantage of allowing the engineer to specify the order of the controller in advance (which is particularly helpful in the reduced order problem), but it completely prevents the determination of optimal order. If one tries to add compensator order as a constraint, a number of problems immediately arise. First of all, how does one express order as a mathematical relationship that can be used in an optimization algorithm? Also, the sizes of the matrices in the Lagrangian and thus the necessary conditions change with changes in order. This gives rise to difficulties in deriving completely general proofs. For example, proofs for full order compensators that require some matrix to be square and full rank may not be valid for higher order compensators because these same matrices may not even be square for the higher order case. Finally, compensator order is not a convex constraint. The mixed H_2/H_∞ problem has a convex objective function subject to a convex constraint [Rid91a,27]. However, if compensator order is added as a constraint, the whole nature of the problem would be changed due to this nonconvex constraint. Therefore, determining the optimal order must be accomplished by discovering some clever parameterization of solutions, completely recasting the problem in some different space, or some form of variational approach.

In order to provide insight into the mixed problem, first consider the optimal order for the related problems of H_2, H_∞, and entropy minimization. The discussion that follows is intended to be more heuristic than rigorous -- the purpose is simply to provide the necessary background of established theory and to lay the foundation for later discussions of the mixed problem.

2.2 H_2 Optimization

This section is divided into two parts. The first part assumes availability of the full state vector for feedback; the other part assumes output feedback. The nature of the solutions turns out to be surprisingly different. Both cases begin with the same system definitions. Consider the H_2 optimization block diagram given in Figure 2-1.

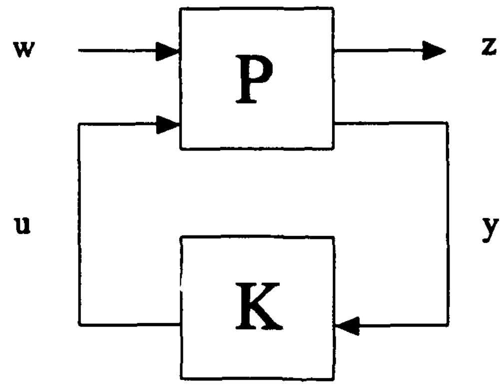

Figure 2-1. H_2 Feedback Control System Block Diagram

Aga

in, the exogenous input w is assumed to be a zero-mean white Gaussian noise of unit intensity. The vector z is the controlled output. The signals u and y are the control inputs and measured outputs, respectively. The state space equations for the generalized transfer matrix P are given by

$$dx/dt = Ax + B_w w + B_u u \tag{2.1a}$$

$$z = C_z x + D_{zw} w + D_{zu} u \tag{2.1b}$$

$$y = C_y x + D_{yw} w + D_{yu} u \tag{2.1c}$$

The transfer function matrix for the open-loop plant P can be partitioned as

$$P = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix}$$

The closed-loop transfer function from w to z is given by the LFT

$$T_{zw} = P_{zw} + P_{zu} K [I - P_{yu}K]^{-1} P_{yw}$$

<u>**2.2.1 State Feedback Case**</u>. The solution is found through a parameterization of the set of all admissible unconstrained H_2-optimal controllers. That is, determine the set of all K(s) such that

$$\| T_{zw}(K) \|_2 \equiv \alpha_o$$

Assume the following conditions on the plant P:

(i) $C_y = I,\ D_{yw} = D_{yu} = 0$ (state feedback)

(ii) (A, B_u) stabilizable

(iii) $D_{zw} = 0$

(iv) $D_{zu}^T D_{zu}$ full rank

(v) $\begin{bmatrix} A - j\omega I & B_u \\ C_z & D_{zu} \end{bmatrix}$ full column rank for all $\omega \in \mathbf{R}$

Let K(s) denote an admissible controller. K(s) can be parameterized by *J* and the freedom parameter *Q* as shown in Figure 2-2.

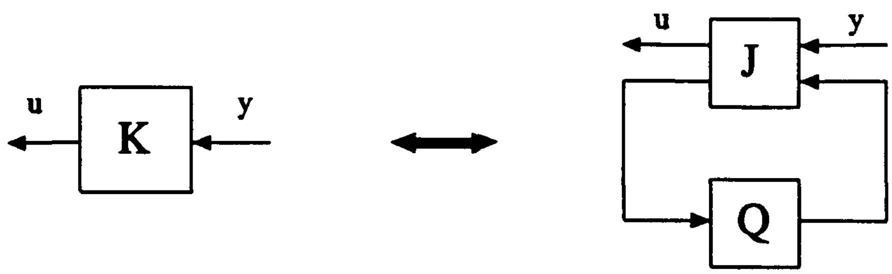

Figure 2-2. Parameterization of K(s)

From Theorem 1 of [RK91], the complete parameterization of optimal controllers, K(s), that minimize $\|T_{zw}\|_2$ are given by the linear fractional transformation of *J* and *Q* with

$$J = \left[\begin{array}{c|cc} A_F & 0 & B_u \\ \hline 0 & F & I \\ -I & I & 0 \end{array}\right]$$

and $Q \in S$, where $S \equiv \{Q \in RH_\infty : Q = W\Pi_1(sI - A_F),\ W \in RH_2\}$

$$F \equiv -(D_{zu}^T D_{zu})^{-1} (D_{zu}^T C_z + B_u^T X)$$

and where X is the unique stabilizing solution to the ARE:

$$A^TX + XA - (D_{zu}^TC_z + B_u^TX)^T(D_{zu}^TD_{zu})^{-1}(D_{zu}^TC_z + B_u^TX) + C_z^TC_z = 0$$

and

$\Pi_1 \equiv I - B_uB_u^+$ (where B_u^+ is the Moore-Penrose pseudoinverse of B_u)

$A_F \equiv A + B_uF$

Notice that when $\mathrm{Im}(B_u) = \mathbf{R}^n$ (or in other words, $B_u \in \mathbf{R}^{n\times m}$, $m \geq n$), this parameterization reduces to a unique, state-feedback (static gain) controller, $K_o = F$. Also, an important result from this parameterization is that when $\mathrm{Im}(B_u)$ is a proper subset of $\mathbf{R}^n$ ($m < n$), there is a family of (dynamic) controllers parameterized by W. Thus, the optimal order under full state availability is zero. Higher order compensators are part of the set of solutions, but they achieve the exact same minimum 2-norm as the static compensator. If the unconstrained problem is the only item of interest, there would be no reason for a designer to choose a higher-order dynamic controller over the static one. However, there is extra freedom provided by this family of H_2-optimal compensators which can be exploited to satisfy some additional constraints. This freedom leads to interesting results in the mixed problem.

Consider briefly the mixed H_2/H_∞ problem (still under full state availability). Rotea & Khargonekar define two problems [RK91,307-308].

Problem A: The true mixed H_2/H_∞ problem -- minimize the H_2-norm, subject to an H_∞ constraint. That is, find

$$\alpha^* \equiv \{ \inf_{K \text{ adm}} \|T_{zw}\|_2, \text{ subj to } \|T_{ed}\|_\infty \leq \gamma\}$$

Problem B: An H_2 super-optimization problem -- determine the unconstrained H_2-optimal controller that also achieves an (added) H_∞ bound. That is, find all K(s) such that

$$\alpha_o \equiv \{ \inf_{K \text{ adm}} \|T_{zw}\|_2\};$$

and the constraint $\|T_{ed}\|_\infty \leq \gamma$ is also trivially satisfied.

Note that a solution to Problem B is also a solution to Problem A (but the reverse is not necessarily true).

Rotea & Khargonekar then provide the necessary and sufficient conditions for the existence of solutions to Problem B (and thus sufficient conditions for existence of solutions to Problem A). Also, under certain conditions, existence of a solution to Problem B is necessary and sufficient for existence of a solution to Problem A. If a solution to Problem B exists, they show that even though the full state is available for feedback, the solution may necessarily be dynamic. This leads them to the conjecture that the optimal order under output feedback is greater than the order of the plant.

Another important thing to notice in Problem B is that the global minimum of $\|T_{zw}\|_2\ (\alpha_o)$ is achieved at each level of $\|T_{ed}\|_\infty$ (if the solution exists). As will be shown, the character of the solution under output feedback is considerably different.

<u>2.2.2 Output Feedback Case</u>. First, parameterize the set of all admissible unconstrained H_2-*sub*optimal controllers. That is, determine the set of all K(s) such that

$$\|T_{zw}(K)\|_2 \leq \alpha, \quad \alpha \geq \alpha_o$$

Assume the following conditions on the plant *P*:

(i) $D_{zw} = 0$

(ii) $D_{yu} = 0$

(iii) (A, B_u) stabilizable, (C_y, A) detectable

(iv) $D_{zu}^T D_{zu}$, $D_{yw} D_{yw}^T$ full rank

Without loss of generality, strengthen this so that

$D_{zu}^T D_{zu} = I$, $D_{yw} D_{yw}^T = I$

(v) $\begin{bmatrix} A - j\omega I & B_u \\ C_z & D_{zu} \end{bmatrix}$ full column rank for all $\omega \in \mathbf{R}$

(vi) $\begin{bmatrix} A-j\omega I & B_w \\ C_y & D_{yw} \end{bmatrix}$ full row rank for all $\omega \in \mathbf{R}$

Note that the case of full state availability is not a trivial special case of output feedback for this development since $D_{yw} = 0$ violates assumption (iv) and leads to a singular control problem.

From [DGKF89], the complete parameterization of all suboptimal admissible controllers, K(s), that achieve $\|T_{zw}\|_2 \leq \alpha$, $\alpha \geq \alpha_0$, is given by the lower LFT of J and Q shown in Figure 2-2, where

$$J = \left[\begin{array}{c|cc} A_J & K_f & K_{f1} \\ \hline -K_c & 0 & I \\ K_{c1} & I & 0 \end{array}\right]$$

$$A_J = A - K_f C_y - B_u K_c$$

$$K_c = B_u^T X_2 + D_{zu}^T C_z, \qquad K_{cl} = -C_y$$

$$K_f = Y_2 C_y^T + B_w D_{yw}^T, \qquad K_{fl} = B_u$$

and where X_2 and Y_2 are the real, unique, symmetric positive semidefinite solutions to the ARE's:

$$(A-B_uD_{zu}^TC_z)^TX_2 + X_2(A-B_uD_{zu}^TC_z) - X_2B_uB_u^TX_2 + [(I-D_{zu}D_{zu}^T)C_z]^T[(I-D_{zu}D_{zu}^T)C_z] = 0$$

$$(A-B_wD_{yw}^TC_y)Y_2 + Y_2(A-B_wD_{yw}^TC_y)^T - Y_2C_y^TC_yY_2 + [B_w(I-D_{yw}^TD_{yw})][B_w(I-D_{yw}^TD_{yw})]^T = 0$$

and

$$Q \in RH_2, \quad \|Q\|_2^2 \leq \alpha^2 - \alpha_o^2$$

If the H_2 optimal compensator is desired (i.e. $\alpha = \alpha_o$), then $Q \equiv 0$, and the solution is unique and has order equal to the order of the plant. Note that this is in contrast to the state feedback case, where a family of optimal compensators exists. If an H_2 suboptimal compensator is desired (i.e. $\alpha > \alpha_o$) in output feedback, then Q(s) is not equal to zero, and a family of suboptimal compensators exists. In summary, under the assumption of output feedback, the optimal compensator is unique and the optimal order is the order of the plant.

Now consider Rotea & Khargonekar's Problem B for the output feedback case. By definition of the problem, $\alpha = \alpha_o$. Therefore, $Q \equiv 0$ and $K(s) \equiv K_{2opt}$ (a unique solution). Define the ∞-norm of T_{ed} when the H_2 optimal compensator is used as $\|T_{ed}(K_{2opt})\|_\infty \equiv \gamma_2$. If a γ is chosen such that $\gamma < \gamma_2$, no solution to Problem B exists at all. If γ is chosen such that $\gamma \geq \gamma_2$, then K(s) is K_{2opt} (the unconstrained H_2-optimal solution). While the family of dynamic H_2 optimal compensators with full state availability provides degrees of freedom that enable γ to be reduced in Problem B, these results are not attainable in the output feedback case because the family of H_2 optimal compensators includes only one unique controller. If Problem B has a solution, it is unique with $\gamma^* = \gamma_2$. Otherwise, there is no solution.

2.3 H_∞ Optimization

Consider the H_∞ optimization block diagram given in Figure 2-3.

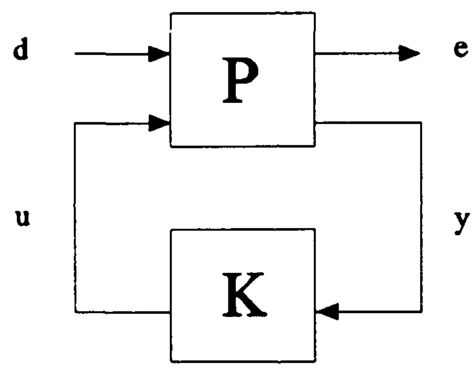

Figure 2-3. H_∞ Feedback Control System Block Diagram

Again, the exogenous input d is assumed to be a deterministic signal of unknown but bounded energy. The vector e is the controlled output. The signals u and y are the control inputs and measured outputs, respectively. The state space equations for the generalized transfer matrix P are given by

$$dx/dt = Ax + B_d d + B_u u \tag{2.2a}$$

$$e = C_e x + D_{ed} d + D_{eu} u \tag{2.2b}$$

$$y = C_y x + D_{yd} d + D_{yu} u \tag{2.2c}$$

The transfer function matrix for the open-loop plant *P* can be partitioned as

$$P = \begin{bmatrix} P_{ed} & P_{eu} \\ P_{yd} & P_{yu} \end{bmatrix}$$

The closed-loop transfer function from *d* to *e* is given by the LFT

$$T_{ed} = P_{ed} + P_{eu} K [I - P_{yu} K]^{-1} P_{yd}$$

Assume the following conditions on the plant *P*:

(i) $D_{ed} = 0$

(ii) $D_{yu} = 0$

(iii) (A, B_u) stabilizable, (C_y, A) detectable

(iv) $D_{eu}^T D_{eu}$, $D_{yd} D_{yd}^T$ full rank

Without loss of generality, strengthen this so that

$D_{eu}^T D_{eu} = I$, $D_{yd} D_{yd}^T = I$

(v) $$\begin{bmatrix} A-j\omega I & B_u \\ C_e & D_{eu} \end{bmatrix}$$ full column rank for all $\omega \in \mathbf{R}$

(vi) $$\begin{bmatrix} A-j\omega I & B_d \\ C_y & D_{yd} \end{bmatrix}$$ full row rank for all $\omega \in \mathbf{R}$

A parameterization of all H_∞ *sub*optimal compensators will now be given. As discussed earlier, the H_∞ optimal solution is not desired and will not be directly addressed here. The problem is to find the family of admissible compensators K(s) such that

$$\|T_{ed}\|_\infty < \gamma, \quad \text{where } \gamma > \gamma_o$$

From [DGKF89], the complete parameterization of suboptimal H_∞ controllers is given by an LFT of *J* and *Q* as shown in the block diagram in Figure 2-2 with

$$J = \left[\begin{array}{c|cc} A_J & K_f & K_{f1} \\ \hline -K_c & 0 & I \\ K_{c1} & I & 0 \end{array}\right]$$

where

$$A_J = A - K_f C_y - B_u K_c + \gamma^{-2} Y_\infty C_e^T (C_e - D_{eu} K_c)$$

$$K_c = (B_u^T X_\infty + D_{eu}^T C_e)(I - \gamma^{-2} Y_\infty X_\infty)^{-1}$$

$$K_f = Y_\infty C_y^T + B_d D_{yd}^T$$

$$K_{cl} = -(\gamma^{-2} D_{yd} B_d^T X_\infty + C_y)(I - \gamma^{-2} Y_\infty X_\infty)^{-1}$$

$$K_{fl} = \gamma^{-2} Y_\infty C_e^T D_{eu} + B_u$$

X_∞ and Y_∞ are the solutions to the ARE's

$$(A - B_u D_{eu}^T C_e)^T X_\infty + X_\infty (A - B_u D_{eu}^T C_e) + X_\infty (\gamma^{-2} B_d B_d^T - B_u B_u^T) X_\infty$$
$$+ C_e^T (I - D_{eu} D_{eu}^T)^T (I - D_{eu} D_{eu}^T) C_e = 0$$

$$(A - B_d D_{yd}^T C_y) Y_\infty + Y_\infty (A - B_d D_{yd}^T C_y)^T + Y_\infty (\gamma^{-2} C_e^T C_e - C_y^T C_y) Y_\infty$$
$$+ B_d (I - D_{yd}^T D_{yd})(I - D_{yd}^T D_{yd})^T B_d^T = 0$$

and the freedom parameter Q is given by

$$Q \in RH_\infty \qquad \|Q\|_\infty < \gamma$$

There are three conditions that must be met in order for this parameterization to be valid:

i) $H_x \in \mathrm{dom(Ric)}$ with $X_\infty = \mathrm{Ric}(H_x) \geq 0$

ii) $H_y \in \mathrm{dom(Ric)}$ with $Y_\infty = \mathrm{Ric}(H_y) \geq 0$

iii) $\rho(Y_\infty X_\infty) < \gamma^2$

where H_x and H_y are the associated Hamiltonian matrices of their respective Riccati equations, and $\rho(Y_\infty X_\infty)$ is the spectral radius of the matrix $Y_\infty X_\infty$. If any of these three conditions are not met, γ is too low (that is, it is below γ_0). This is where the iterative nature of H_∞ optimization is seen. γ_0 must be approached iteratively, and γ can be made to be arbitrarily close to the optimal value.

Note that the simplest controller in this parameterization is given when $Q=0$. This is known as the central H_∞ compensator. In this case, the compensator is unique and its order is equal to the order of the plant. Other higher order compensators are in the set of solutions (for nonzero values of Q), but the maximum order required for this problem is the order of the plant.

Without rigorous development, consider briefly the H_∞ optimal case. When γ gets close to γ_0, the H_∞ suboptimal controller (at least in a SISO problem) has a singular value plot that is completely flat from low frequency out to a high frequency roll-off. In other words, it looks like an all pass filter with a single high frequency pole. As γ approaches γ_0, this pole moves toward infinity. At H_∞ optimal (that is, $\gamma = \gamma_0$), the pole is at infinity and the controller actually drops rank [Rid91b]. Therefore, the optimal order for H_∞ optimal compensators is no greater than full order minus one.

2.4 Minimum Entropy

The concept of the entropy of a control system is very non-intuitive and is not directly related to the traditional ideas of thermodynamic randomness. Let $G \in RL_\infty$ and $\|G\|_\infty < \gamma$. The entropy at infinity of G(s) is defined by

$$I[G(s),\gamma] \equiv \lim_{s_o \to \infty} \left\{ -\frac{\gamma^2}{2\pi} \int_{-\infty}^{\infty} \ln|\det[I - \gamma^{-2} G(j\omega)^* G(j\omega)]| \left[\frac{s_o^2}{s_o^2+\omega^2}\right] d\omega \right\}$$

Note from examination of this definition that the entropy has the following properties:

i) $I[G(s),\gamma] \geq 0$

ii) $I[G(s),\gamma] = 0$ iff $G(s) \equiv 0$

iii) $I[G(s),\gamma] < \infty$ iff $G(\infty) = 0$ [MG90,8-11]

While this definition does not appear to have much practical usefulness in this form, entropy is a useful quantity and has some interesting relationships to the 2-norm and ∞-norm. In particular, entropy is an upper bound to the 2-norm and is equal to 2-norm as γ gets very large. That is, if $G \in RH_2$ and γ is such that $\|G\|_\infty < \gamma$, then $I[G(s),\gamma] \geq \|G(s)\|_2^2$ and $I[G(s),\infty] = \|G(s)\|_2^2$ ([MG90,12], Theorem 2.4.4). Also, entropy is relatively easy to compute in state space despite its formidable definition. Let $G \in RH_2$, $\|G\|_\infty < \gamma$, and $G(s) = C(sI-A)^{-1}B$. Then

$$I[G(s),\gamma] = \text{tr}\,[QC^TC]$$

where $Q = Q^T > 0$ is the stabilizing solution to the ARE

$$0 = AQ + QA^T + \gamma^{-2}QC^TCQ + BB^T \qquad \text{[MG90,52-54] Lemma 5.3.2}$$

Now, consider minimizing the closed-loop entropy of a system. Make all the same assumptions that were made for the H_∞ suboptimal parameterization and let $\gamma > \gamma_o$. Then, minimizing the closed-loop entropy $I[T_{ed}(s),\gamma]$ over the set of all admissible compensators K(s) such that $\|T_{ed}\|_\infty < \gamma$ results in the central H_∞ controller at that γ [MG90]. The order of the central H_∞ controller has already been shown to be no greater than the order of the plant. Therefore, the minimum entropy problem has an optimal order that is no greater than the order of the plant.

Table 2-1 gives a summary of the optimal orders of the compensators that are solutions to the optimization problems just discussed.

Table 2-1. Optimal Compensator Orders

Optimization Problem	Optimal Order of Compensator
H_2 (opt)	0 (full-state feedback)
	$\leq n$ (output feedback)
H_∞ (sub-opt)	$\leq n$
H_∞ (opt)	$\leq n-1$
$I\,[T_{ed}, \gamma]$	$\leq n$

note: n = order of plant

III. General Development of the Mixed H_2/H_∞ Problem

3.1 Problem Statement

Consider again the general form of the linear time-invariant feedback control system for the mixed H_2/H_∞ optimization problem as introduced in Chapter 1, reproduced in Figure 3-1.

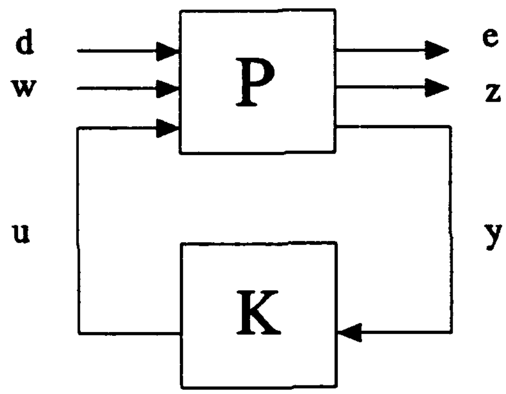

Figure 3-1. Mixed H_2/H_∞ System Block Diagram

The exogenous input d is assumed to be a deterministic signal of unknown but bounded energy. The input w is assumed to be a zero-mean white Gaussian noise of unit intensity. The signals e and z are controlled outputs, and may be equal, dependent, or independent. The signals u and y are the control inputs and measured outputs, respectively. All weighting functions required to obtain this general form are included in the generalized plant transfer function P. The state space equations for P are given by:

$$dx/dt = Ax + B_d d + B_w w + B_u u \tag{3.1a}$$

$$e = C_e x + D_{ed} d + D_{ew} w + D_{eu} u \tag{3.1b}$$

$$z = C_z x + D_{zd} d + D_{zw} w + D_{zu} u \tag{3.1c}$$

$$y = C_y x + D_{yd} d + D_{yw} w + D_{yu} u \tag{3.1d}$$

The transfer function matrix for the open-loop plant P can be partitioned as

$$P = \begin{bmatrix} P_{ed} & P_{ew} & P_{eu} \\ P_{zd} & P_{zw} & P_{zu} \\ P_{yd} & P_{yw} & P_{yu} \end{bmatrix}$$

Closed-loop transfer functions from w to z and from d to e are given by the lower linear fractional transformations

$$T_{zw} = P_{zw} + P_{zu}\, K\, [I-P_{yu}K]^{-1}\, P_{yw}$$

$$T_{ed} = P_{ed} + P_{eu}\, K\, [I-P_{yu}K]^{-1}\, P_{yd}$$

Now, assume the plant P satisfies the following conditions:

(i) $D_{zw} = 0$

(ii) $D_{ed} = 0$

(iii) $D_{yu} = 0$

(iv) (A,B_u) stabilizable, (C_y,A) detectable

(v) $D_{zu}^T D_{zu}$, $D_{yw} D_{yw}^T$ full rank

(vi) $D_{eu}^T D_{eu}$, $D_{yd} D_{yd}^T$ full rank

(vii) $\begin{bmatrix} A-j\omega I & B_u \\ C_z & D_{zu} \end{bmatrix}$ full column rank for all $\omega \in \mathbf{R}$

(viii) $\begin{bmatrix} A-j\omega I & B_w \\ C_y & D_{yw} \end{bmatrix}$ full row rank for all $\omega \in \mathbf{R}$

(ix) $\begin{bmatrix} A-j\omega I & B_u \\ C_e & D_{eu} \end{bmatrix}$ full column rank for all $\omega \in \mathbf{R}$

(x) $\begin{bmatrix} A-j\omega I & B_d \\ C_y & D_{yd} \end{bmatrix}$ full row rank for all $\omega \in \mathbf{R}$

The rationale for these assumptions are:

i) Required for a well-posed H_2 problem. If $D_{zw} \neq 0$, the 2-norm of T_{zw} will be infinite, regardless of K.

ii) Not required for finite ∞-norm, but greatly simplifies the development.

iii) Not required, but greatly simplifies the development and is representative of physically realizable plants.

iv) Necessary for the existence of any stabilizing controller.

v) - x) Ensures the pure H_2 and H_∞ optimization problems each have an admissible solution.

The following definitions will be used throughout:

$$\gamma_o \equiv \inf_{K \text{ adm}} \|T_{ed}\|_\infty$$

$$\alpha_o \equiv \inf_{K \text{ adm}} \|T_{zw}\|_2$$

K_{2opt} ≡ unique K(s) that gives $\|T_{zw}\|_2 = \alpha_o$

γ_2 ≡ $\|T_{ed}\|_\infty$ when $K(s) = K_{2opt}$

K_{mix} ≡ a K(s) that solves the mixed H_2/H_∞ problem at some γ

γ^* ≡ $\|T_{ed}\|_\infty$ when $K(s) = K_{mix}$

α^* ≡ $\|T_{zw}\|_2$ when $K(s) = K_{mix}$

n ≡ order of the plant

n_c ≡ order of the compensator

n^* ≡ optimal order of K_{mix}

Before getting too deep into the development of the mixed problem, briefly survey the big picture. Consider plotting the 2-norm of T_{zw} versus the ∞-norm of T_{ed}. Figure 3-2 shows a generic plot with some key points identified.

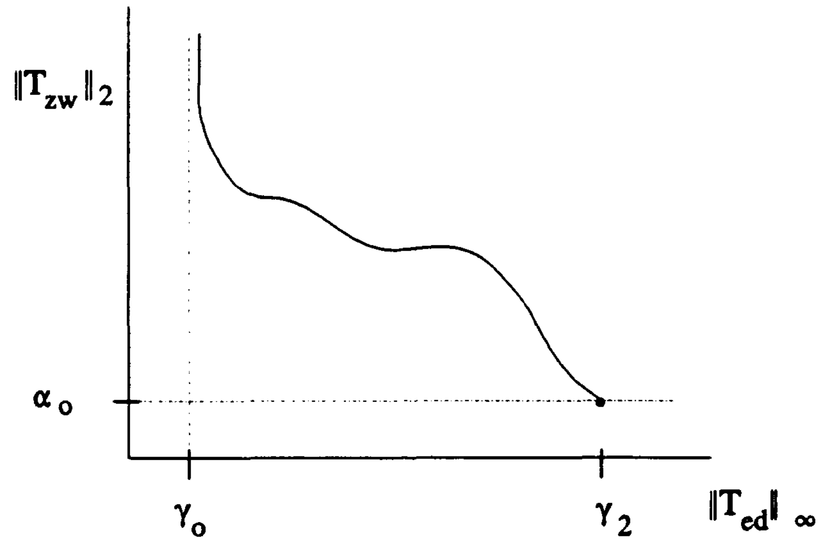

Figure 3-2. Generic H_2 versus H_∞ Plot

This plot serves to illustrate the boundaries of the mixed problem. Note that α_o is the lowest achievable 2-norm by any compensator since it comes from the unconstrained H_2 problem. Likewise γ_o is the minimum achievable ∞-norm. Therefore, the dashed lines represent limits of achievable H_2 and H_∞ performance within which the mixed solutions must lie. If γ is lowered beyond γ_o, no solution will exist. Note also that as the H_∞ optimal solution is approached, the 2-norm becomes infinite. Actually, there are possible cases where the optimal H_∞ controller has a zero state space D matrix, thus allowing a finite 2-norm. These cases are rare and will not be considered, even though the development to follow would still hold. Finally, as γ is increased to γ_2, the solution becomes the unique H_2 unconstrained optimal solution. The solution will remain the same unique H_2 optimal solution if γ is increased further because the ∞-norm constraint is effectively removed.

Now, the general mixed H_2/H_∞ optimization synthesis problem is: Determine the set of admissible compensators K(s) such that

$$\inf_{K\ adm} \|T_{zw}\|_2, \quad \text{subject to the constraint} \quad \|T_{ed}\|_\infty \leq \gamma$$

is achieved.

First, this problem statement needs to be turned into a mathematical statement that can be manipulated. Begin with a definition of the compensator. A state space description of the compensator in Figure 3-1 is

$$dx_c/dt = A_c x_c + B_c y \tag{3.2a}$$

$$u = C_c x_c + D_c y \tag{3.2b}$$

It is easily shown that D_c must be zero in order to have a finite T_{zw} [Rid91a,92], so the assumption that K(s) is strictly proper is made with no loss of generality.

Using the control law $u = K(s)y$, form the closed-loop system with the augmented state vector $\underline{x} = \begin{bmatrix} x \\ x_c \end{bmatrix}$ by combining Equations (3.1) and (3.2).

The closed-loop system can now be written in state space form as

$$\frac{d\tilde{x}}{dt} = \tilde{A}\tilde{x} + \tilde{B}_d d + \tilde{B}_w w \tag{3.3a}$$

$$e = \tilde{C}_e\tilde{x} + D_{ew} w \tag{3.3b}$$

$$z = \tilde{C}_z\tilde{x} + D_{zd} d \tag{3.3c}$$

where the closed-loop (tilde) matrices are given by

$$\tilde{A} = \begin{bmatrix} A & B_u C_c \\ B_c C_y & A_c \end{bmatrix}$$

$$\tilde{B}_d = \begin{bmatrix} B_d \\ B_c D_{yd} \end{bmatrix} \qquad \tilde{B}_w = \begin{bmatrix} B_w \\ B_c D_{yw} \end{bmatrix}$$

$$\tilde{C}_e = [\, C_e \quad D_{eu} C_c \,] \qquad \tilde{C}_z = [\, C_z \quad D_{zu} C_c \,]$$

The closed-loop transfer functions from w to z and from d to e can now be written as

$$T_{ed} = \tilde{C}_e \, (sI - \tilde{A})^{-1} \, \tilde{B}_d$$

$$T_{zw} = \tilde{C}_z \, (sI - \tilde{A})^{-1} \, \tilde{B}_w$$

Now that the closed-loop system has been expressed in terms of the unknown compensator matrices (A_c, B_c, C_c), the cost objective, stability requirement, and constraint need to be recast as mathematical functions. Begin with the cost objective and stability requirement. It is a standard result that the 2-norm squared of a transfer function can be found as a function of the solution to a Lyapunov equation. Specifically, if

$$T_{zw}(s) = \left[\begin{array}{c|c} \tilde{A} & \tilde{B}_w \\ \hline \tilde{C}_z & 0 \end{array}\right] \in RH_2$$

(which is true with the given assumptions)

then

$$\|T_{zw}\|_2^2 = \mathrm{tr}\,[\tilde{Q}_2\tilde{C}_z^T\tilde{C}_z]$$

where $\tilde{Q}_2 = \tilde{Q}_2^T \geq 0$ is the solution to the Lyapunov equation

$$\tilde{A}\tilde{Q}_2 + \tilde{Q}_2\tilde{A}^T + \tilde{B}_w\tilde{B}_w^T = 0$$

Note from Lyapunov theory that this symmetric, positive semidefinite solution only exists when $\tilde{A}$ is stable [Won85,283], so the requirement of finding a stabilizing compensator will be automatically satisfied if the solution of this equation is enforced.

Now consider the ∞-norm constraint. It can be shown that an ∞-norm bound on a transfer function can be guaranteed by requiring the solution to a Riccati equation. Specifically, for

$$T_{ed}(s) = \left[\begin{array}{c|c} \tilde{A} & \tilde{B}_d \\ \hline \tilde{C}_e & 0 \end{array}\right]$$

and assuming $\tilde{A}$ is stable (which is enforced by the Lyapunov equation), then if there exists a $Q_\infty = Q_\infty^T \geq 0$ that satisfies the algebraic Riccati equation

$$\tilde{A}Q_\infty + Q_\infty\tilde{A}^T + \gamma^{-2}Q_\infty\tilde{C}_e^T\tilde{C}_e Q_\infty + \tilde{B}_d\tilde{B}_d^T = 0$$

then

$$\| T_{ed} \|_\infty \leq \gamma \qquad \text{[Rid91a,55-57] Thm 2.5.17}$$

Finally, the mixed H_2/H_∞ problem can now be restated as:

Determine (A_c, B_c, C_c) that minimizes the cost function

$$J(A_c, B_c, C_c) = \text{tr}\,[\tilde{Q}_2\tilde{C}_z^T\tilde{C}_z] \tag{3.4}$$

where $\tilde{Q}_2$ is the real, symmetric, positive semidefinite solution to

$$\tilde{A}\tilde{Q}_2 + \tilde{Q}_2\tilde{A}^T + \tilde{B}_w\tilde{B}_w^T = 0$$

and such that (for a given γ)

$$\tilde{A}Q_\infty + Q_\infty \tilde{A}^T + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e Q_\infty + \tilde{B}_d \tilde{B}_d^T = 0$$

has a real, symmetric, positive semidefinite solution.

Note that the variables in the problem are A_c, B_c, C_c, $\tilde{Q}_2$, and Q_∞. All other matrices are known from the system model, and γ is given in the problem statement.

3.2 First-Order Necessary Conditions for the General Mixed Problem

In order to derive the necessary conditions for the mixed H_2/H_∞ optimization problem, the method of Lagrange multipliers is used. In this method, a constrained optimization problem is transformed into an unconstrained problem by adjoining the constraint equations to the original cost function and forming a new cost function, called the Lagrangian. The first variations of this Lagrangian are then set equal to zero to find the necessary conditions for a minimum (second variations give the sufficient conditions). The constraint equations (which must be of the form $f(x,y,...) = 0$) are adjoined by a parameter called the Lagrange multiplier. This Lagrange multiplier is then another variable which must be solved for. If, during the solution of the problem, it is shown that the Lagrange multiplier must be zero, it physically means that the constraint is not in effect and can be removed.

For this problem, there are two constraint equations to be adjoined. Since they are matrix equations, the Lagrange multipliers must also be matrices. Define these two matrices in partitioned form as

$$X = \begin{bmatrix} X_1 & X_{12} \\ X_{12}^T & X_2 \end{bmatrix} \qquad Y = \begin{bmatrix} Y_1 & Y_{12} \\ Y_{12}^T & Y_2 \end{bmatrix}$$

These matrices do not have to be defined as symmetric. However, they will be shown to be real symmetric solutions to Lyapunov equations, so for simplicity they will be defined to be symmetric here (X_1, X_2, Y_1, and Y_2 are therefore symmetric).

The Lagrangian can now be formed and the mixed H_2/H_∞ Lagrange multiplier problem can be stated. Assume $\tilde{Q}_2 = \tilde{Q}_2^T \geq 0$ and $Q_\infty = Q_\infty^T \geq 0$. Minimize the Lagrangian

$$\begin{aligned} \mathscr{L} = {} & \mathrm{tr}[\tilde{Q}_2 \tilde{C}_z^T \tilde{C}_z] + \mathrm{tr}\{[\tilde{A}\tilde{Q}_2 + \tilde{Q}_2\tilde{A}^T + \tilde{B}_w\tilde{B}_w^T]X\} \\ & + \mathrm{tr}\{[\tilde{A}Q_\infty + Q_\infty\tilde{A}^T + \gamma^{-2}Q_\infty\tilde{C}_e^T\tilde{C}_eQ_\infty + \tilde{B}_d\tilde{B}_d^T]Y\} \end{aligned} \tag{3.5}$$

Note that if $Y = 0$, the ∞-norm bound is trivially satisfied. That is, the constraint is inactive. If the solution lies on the boundary of this constraint, then $Y \neq 0$. For the special case $\gamma = \gamma_2$, the solution lies on the boundary <u>and</u> $Y \equiv 0$.

Now set the first variations of this function equal to zero. That is, take the partial derivatives of $\mathscr{L}$ with respect to all the variables (A_c, B_c, C_c, $\tilde{Q}_2$, Q_∞, X, and Y) and set them equal to zero. These are derivatives of scalars with respect to matrices for which the following formulae apply:

$$\frac{\partial \, tr(AXB)}{\partial X} = A^T B^T$$

$$\frac{\partial \, tr(AX^T B)}{\partial X} = B A$$

$$\frac{\partial \, tr(AXBXC)}{\partial X} = A^T C^T X^T B^T + B^T X^T A^T C^T$$

$$\frac{\partial \, tr(AXBX^T C)}{\partial X} = A^T C^T X B^T + CAXB$$

[AS65]

The derivatives with respect to $\tilde{Q}_2$, Q_∞, X, and Y are no problem because they appear explicitly in the Lagrangian in its present form. However, the compensator matrices A_c, B_c, and C_c do not appear explicitly, so the Lagrangian must be multiplied out and expanded into its constituent sub-blocks. Define the sub-blocks of $\tilde{Q}_2$ and Q_∞ as (note: these are symmetric matrices)

$$\tilde{Q}_2 = \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \qquad Q_\infty = \begin{bmatrix} Q_a & Q_{ab} \\ Q_{ab}^T & Q_b \end{bmatrix}$$

The off-diagonal terms in these partitions are not square unless the order of the compensator is equal to the order of the plant. In general, the sizes of the matrices within the partitions of $\tilde{A}$, $\tilde{Q}_2$, Q_∞, X, and Y are

$$\begin{bmatrix} \\ (n+n_c) \times (n+n_c) \\ \\ \end{bmatrix} = \left[\begin{array}{c:c} n \times n & n \times n_c \\ \hdashline n_c \times n & n_c \times n_c \end{array}\right]$$

In addition, to reduce excessive notation, make the following simplifying definitions:

$$V_a = B_d B_d^T \qquad V_1 = B_w B_w^T$$

$$V_{ab} = B_d D_{yd}^T \qquad V_{12} = B_w D_{yw}^T$$

$$V_b = D_{yd} D_{yd}^T \qquad V_2 = D_{yw} D_{yw}^T$$

$$R_a = C_e^T C_e \qquad R_1 = C_z^T C_z$$

$$R = C_e^T D_{eu} \qquad R_{12} = C_z^T D_{zu}$$

$$R_b = D_{eu}^T D_{eu} \qquad R_2 = D_{zu}^T D_{zu}$$

After substituting all the expressions for the closed-loop system, $\tilde{Q}_2$, Q_∞, X, Y, the compensator, and the simplifying definitions, the expanded Lagrangian is

$$
\begin{aligned}
\mathscr{L} = \mathrm{tr}\{\ & Q_1R_1 + Q_{12}C_c^T R_{12}^T + Q_{12}^T R_{12}C_c + Q_2C_c^T R_2C_c + AQ_1X_1 \\
& + Q_1A^TX_1 + B_uC_cQ_{12}^TX_1 + Q_{12}C_c^T B_u^T X_1 + V_1X_1 + AQ_{12}X_{12}^T + Q_{12}A_c^T X_{12}^T \\
& + B_uC_cQ_2X_{12}^T + Q_1C_y^T B_c^T X_{12}^T + V_{12}B_c^T X_{12}^T + B_cC_yQ_1X_{12} \\
& + A_cQ_{12}^TX_{12} + Q_{12}^TA^TX_{12} + Q_2C_c^T B_u^T X_{12} + B_cV_{12}^TX_{12} + A_cQ_2X_2 \\
& + Q_2A_c^T X_2 + B_cC_yQ_{12}X_2 + Q_{12}^TC_y^T B_c^T X_2 + B_cV_2B_c^T X_2 + AQ_aY_1 \\
& + Q_aA^TY_1 + B_uC_cQ^TY_1 + Q_{ab}C_c^T B_u^T Y_1 + V_aY_1 + AQ_{ab}Y_{12}^T + Q_{ab}A_c^T Y_{12}^T \\
& + B_uC_cQ_bY_{12}^T + Q_aC_y^T B_c^T Y_{12}^T + V_{ab}B_c^T Y_{12}^T + B_cC_yQ_aY_{12} + A_cQ_{ab}^TY_{12} \\
& + Q_{ab}^TA^TY_{12} + Q_bC_c^T B_u^T Y_{12} + B_cV_{ab}^TY_{12} + A_cQ_bY_2 + Q_bA_c^T Y_2 \\
& + B_cC_yQ_{ab}Y_2 + Q_{ab}^TC_y^T B_c^T Y_2 + B_cV_2B_c^T Y_2 + \gamma^{-2}[\ Q_aR_aQ_aY_1 \\
& + Q_{ab}C_c^T R_{ab}^TQ_aY_1 + Q_aR_{ab}C_cQ_{ab}^TY_1 + Q_{ab}C_c^T R_bC_cQ_{ab}^TY_1 + Q_aR_aQ_{ab}Y_{12}^T \\
& + Q_{ab}C_c^T R_{ab}^TQ_{ab}Y_{12}^T + Q_aR_{ab}C_cQ_bY_{12}^T + Q_{ab}C_c^T R_bC_cQ_bY_{12}^T \\
& + Q_{ab}^TR_aQ_aY_{12} + Q_bC_c^T R_{ab}^TQ_aY_{12} + Q_{ab}^TR_{ab}C_cQ_{ab}^TY_{12} \\
& + Q_bC_c^T R_bC_cQ_{ab}^TY_{12} + Q_{ab}^TR_aQ_{ab}Y_2 + Q_bC_c^T R_{ab}^TQ_{ab}Y_2 \\
& + Q_{ab}^TR_{ab}C_cQ_bY_2 + Q_bC_c^T R_bC_cQ_bY_2]\ \}
\end{aligned}
\tag{3.6}
$$

Now, the partial derivatives of the Lagrangian are [Rid91a,97-98]

$$\frac{\partial \mathcal{L}}{\partial A_c} = X_{12}^T Q_{12} + X_2 Q_2 + Y_{12}^T Q_{ab} + Y_2 Q_b = 0 \tag{3.7}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial B_c} = \; & X_{12}^T Q_1 C_y^T + X_2 Q_{12}^T C_y^T + X_{12}^T V_{12} + X_2 B_c V_2 + Y_{12}^T Q_a C_y^T \\ & + Y_2 Q_{ab}^T C_y^T + Y_{12}^T V_{ab} + Y_2 B_c V_b = 0\end{aligned} \tag{3.8}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial C_c} = \; & B_u^T X_1 Q_{12} + B_u^T X_{12} Q_2 + R_{12}^T Q_{12} + R_2 C_c Q_2 + B_u^T Y_1 Q_{ab} \\ & + B_u^T Y_{12} Q_b + \gamma^{-2} [\, R_{ab}^T Q_a Y_1 Q_{ab} + R_{ab}^T Q_a Y_{12} Q_b \\ & + R_{ab}^T Q_{ab} Y_{12}^T Q_{ab} + R_{ab}^T Q_{ab} Y_2 Q_b + R_b C_c Q_{ab}^T Y_1 Q_{ab} \\ & + R_b C_c Q_b Y_{12}^T Q_{ab} + R_b C_c Q_{ab}^T Y_{12} Q_b + R_b C_c Q_b Y_2 Q_b \,] = 0\end{aligned} \tag{3.9}$$

$$\frac{\partial \mathcal{L}}{\partial X} = \tilde{A}\tilde{Q}_2 + \tilde{Q}_2 \tilde{A}^T + \tilde{B}_w \tilde{B}_w^T = 0 \tag{3.10}$$

$$\frac{\partial \mathcal{L}}{\partial \tilde{Q}_2} = \tilde{A}^T X + X\tilde{A} + \tilde{C}_z^T \tilde{C}_z = 0 \tag{3.11}$$

$$\frac{\partial \mathcal{L}}{\partial Y} = \tilde{A} Q_\infty + Q_\infty \tilde{A}^T + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e Q_\infty + \tilde{B}_d \tilde{B}_d^T = 0 \tag{3.12}$$

$$\frac{\partial \mathcal{L}}{\partial Q_\infty} = [\tilde{A} + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e]^T\, Y + Y\,[\tilde{A} + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e] = 0 \tag{3.13}$$

These are a set of seven nonlinear, coupled matrix equations for which no analytical solution has been found. Equations (3.10) and (3.11) are Lyapunov equations in $\tilde{Q}_2$ and X, respectively. It is interesting to note that these two equations show that $\tilde{Q}_2$ and X are the controllability and observability gramians of T_{zw}. Equation (3.13) is a Lyapunov equation in Y, but it has no constant term. Therefore, if the matrix $(\tilde{A} + \gamma^{-2}Q_\infty\tilde{C}_e^T\tilde{C}_e)$ is stable, the only solution to (3.13) is Y = 0 ([SZ70], Thm 2.1). Now, Equation (3.12) is a Riccati equation in Q_∞. If Q_∞ is chosen to be the stabilizing solution to this Riccati equation (which is typically the solution that is wanted), the matrix that is stabilized is $(\tilde{A} + \gamma^{-2}Q_\infty\tilde{C}_e^T\tilde{C}_e)$. This means that the Lagrange multiplier matrix Y must be equal to zero, which means that the ∞-norm constraint is not active and can effectively be removed.

This apparent contradiction might lead one to question the initial set-up of the problem. However, one of the key contributions of Ridgely's work was the recognition that the neutrally stabilizing solution (not the stabilizing solution) to (3.12) is the solution that is desired in order to obtain an on-boundary mixed solution. If $(\tilde{A} + \gamma^{-2}Q_\infty\tilde{C}_e^T\tilde{C}_e)$ is neutrally stable (that is, it has at least one $j\omega$-axis eigenvalue with the remaining eigenvalues in the left half plane), then there are an infinite number of non-zero solutions (Y) of varying rank ([SZ70], Lemmas 4.1 and 4.2). This immediately brings up the question of uniqueness of the mixed H_2/H_∞ solution. No proofs of uniqueness have been given, and although this issue was touched upon in this research while trying to prove optimal order, no proofs of uniqueness will be offered here. While it may turn out to be true that the solution is indeed unique, it would not be surprising to find out that it is not, since both the H_2-suboptimal and H_∞-suboptimal (and optimal) are non-unique.

Since these equations are intractable analytically, they must be solved numerically (Note: even though closed-form solutions for A_c, B_c, C_c are not available, some important results can be shown from these equations. Some of the key theorems that describe the nature of the mixed problem for full order controllers are given later in Section 3.4). There are software programs readily available that solve Riccati and Lyapunov equations. However, they do not return neutrally stabilizing solutions to Riccati equations and do not handle Lyapunov equations that have no constant terms. The equations could be numerically handled without using any Riccati or Lyapunov solvers, but the number of unknowns becomes unreasonable very quickly. Also, the numerics become erratic near the solution because the solution lies just on the boundary of non-existence of solutions. Therefore, even to solve this problem numerically, modifications are needed. This is the motivation for setting up the *sub*optimal mixed H_2/H_∞ problem, described next.

3.3 First-Order Necessary Conditions for the Suboptimal Mixed Problem

The objective now is to develop a suboptimal mixed H_2/H_∞ problem that has a set of necessary conditions that are more numerically stable and that approach the optimal solution. Recall the statement of general mixed problem:

Determine (A_c, B_c, C_c) that minimizes the cost function

$$J(A_c, B_c, C_c) = \text{tr}\,[\tilde{Q}_2 \tilde{C}_z^T \tilde{C}_z]$$

where $\tilde{Q}_2$ is the real, symmetric, positive semidefinite solution to

$$\tilde{A}\tilde{Q}_2 + \tilde{Q}_2\tilde{A}^T + \tilde{B}_w\tilde{B}_w^T = 0$$

and such that

$$\tilde{A}Q_\infty + Q_\infty\tilde{A}^T + \gamma^{-2}Q_\infty\tilde{C}_e^T\tilde{C}_eQ_\infty + \tilde{B}_d\tilde{B}_d^T = 0$$

has a real, symmetric, positive semidefinite solution.

Now, for the suboptimal mixed problem, define a new cost objective (the constraint equations remain unchanged):

$$J_\mu(A_c, B_c, C_c) = (1-\mu)\,\text{tr}[\tilde{Q}_2\tilde{C}_z^T\tilde{C}_z] + \mu\,\text{tr}[Q_\infty\tilde{C}_e^T\tilde{C}_e] \tag{3.14}$$

where

$\mu \in \mathbf{R}, \quad \mu \in [0,1]$

Notice that when $\mu = 0$, the problem reduces to the original optimal mixed problem. The additional term, $Q_\infty(\tilde{C}_e^T \tilde{C}_e)$ in the cost function is partially variable as $(\tilde{C}_e^T \tilde{C}_e)$ could actually be any real, symmetric, positive semidefinite matrix. The reason for choosing $(\tilde{C}_e^T \tilde{C}_e)$ is that, for $\mu \neq 0$, the term $\mathrm{tr}[Q_\infty \tilde{C}_e^T \tilde{C}_e]$ is the entropy of T_{ed} ([Rid91a,118], Thm 5.1.1). This is very convenient because it provides an excellent starting point for the numerical algorithm. When $\mu = 1$, the problem reduces to pure entropy minimization for which a closed-form solution is available (it is the central H_∞ controller at the given level of γ).

The reason for adding the extra term into the cost function was to try to modify the necessary conditions so that the Lyapunov equation in the necessary conditions, Equation (3.13), would have a small symmetric positive semidefinite constant term. This would enable the use of a Lyapunov solver and would require the term $(\tilde{A} + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e)$ to be stable without $Y \equiv 0$. Hence, the stabilizing solution to Equation (3.12) would be desired and a Riccati solver could also be used. The additional term does indeed accomplish this.

With the new cost function defined, the suboptimal mixed H_2/H_∞ Lagrangian becomes

$$
\begin{aligned}
\mathscr{L}_\mu = {} & (1-\mu)\, \mathrm{tr}[\tilde{Q}_2 \tilde{C}_z^T \tilde{C}_z] + \mu\, \mathrm{tr}[Q_\infty \tilde{C}_e^T \tilde{C}_e] \\
& + \mathrm{tr}\{[\tilde{A}\tilde{Q}_2 + \tilde{Q}_2 \tilde{A}^T + \tilde{B}_w \tilde{B}_w^T]X\} \\
& + \mathrm{tr}\{[\tilde{A} Q_\infty + Q_\infty \tilde{A}^T + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e Q_\infty + \tilde{B}_d \tilde{B}_d^T]Y\}
\end{aligned}
\tag{3.15}
$$

Now, the partial derivatives of this new Lagrangian are [Rid91a,116-117]

$$\frac{\partial \mathscr{L}_\mu}{\partial A_c} = X_{12}^T Q_{12} + X_2 Q_2 + Y_{12}^T Q_{ab} + Y_2 Q_b = 0 \tag{3.16}$$

$$\frac{\partial \mathscr{L}_\mu}{\partial B_c} = X_{12}^T Q_1 C_y^T + X_2 Q_{12}^T C_y^T + X_{12}^T V_{12} + X_2 B_c V_2 + Y_{12}^T Q_a C_y^T$$
$$+ Y_2 Q_{ab}^T C_y^T + Y_{12}^T V_{ab} + Y_2 B_c V_b = 0 \tag{3.17}$$

$$\frac{\partial \mathscr{L}_\mu}{\partial C_c} = B_u^T X_1 Q_{12} + B_u^T X_{12} Q_2 + (1-\mu) R_{12}^T Q_{12} + (1-\mu) R_2 C_c Q_2$$
$$+ B_u^T Y_1 Q_{ab} + B_u^T Y_{12} Q_b + \mu R_{ab}^T Q_{ab} + \mu R_b C_c Q_b$$
$$+ \gamma^{-2} [R_{ab}^T Q_a Y_1 Q_{ab} + R_{ab}^T Q_a Y_{12} Q_b + R_{ab}^T Q_{ab} Y_{12}^T Q_{ab}$$
$$+ R_{ab}^T Q_{ab} Y_2 Q_b + R_b C_c Q_{ab}^T Y_1 Q_{ab} + R_b C_c Q_b Y_{12}^T Q_{ab}$$
$$+ R_b C_c Q_{ab}^T Y_{12} Q_b + R_b C_c Q_b Y_2 Q_b] = 0 \tag{3.18}$$

$$\frac{\partial \mathscr{L}_\mu}{\partial X} = \tilde{A}\tilde{Q}_2 + \tilde{Q}_2 \tilde{A}^T + \tilde{B}_w \tilde{B}_w^T = 0 \tag{3.19}$$

$$\frac{\partial \mathscr{L}_\mu}{\partial \tilde{Q}_2} = \tilde{A}^T X + X\tilde{A} + (1-\mu)\tilde{C}_z^T \tilde{C}_z = 0 \tag{3.20}$$

$$\frac{\partial \mathscr{L}_\mu}{\partial Y} = \tilde{A} Q_\infty + Q_\infty \tilde{A}^T + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e Q_\infty + \tilde{B}_d \tilde{B}_d^T = 0 \tag{3.21}$$

$$\frac{\partial \mathscr{L}_\mu}{\partial Q_\infty} = [\tilde{A} + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e]^T Y + Y[\tilde{A} + \gamma^{-2} Q_\infty \tilde{C}_e^T \tilde{C}_e] + \mu \tilde{C}_e^T \tilde{C}_e = 0 \tag{3.22}$$

It is immediately obvious that the clever modification of the cost function is very helpful for setting up a more well-behaved problem for numerical solution. Equation (3.22) is a typical Lyapunov equation, and Equation (3.21) now requires a stabilizing solution. At $\mu = 0$, the necessary conditions are exactly the same as the ones derived for the optimal mixed problem. At $\mu = 1$, the problem is the minimum entropy problem for which a closed-form solution can be found. Now, all that is required in the numerical solution is to start with the central H_∞ controller (with $\mu = 1$) and successively reduce μ until the solution converges to the optimal mixed solution (which it does, as will be shown in the next section).

3.4 Summary of Full Order Results

The suboptimal mixed problem is set up purely for the sake of performing the numerical solution. Even though the optimal mixed problem could not be solved in closed-form, some important characteristics of the solutions have been proven for compensators of equal order to the plant. The key proofs that relate directly to the increased order problem are summarized here.

Notice that until now, the order of the compensator K(s) has not been specified, so all the necessary conditions are valid for any order. However, as compensator order changes, the sizes of the equations change. This means that in order to continue with the analysis (either numerically or analytically) the order of the compensator must be chosen. In Ridgely's work, the order was selected to be full order (that is, $n_c = n$). The majority of the proofs and examples given in this section, therefore, are given under the assumption of a full order compensator.

Theorem 3.4.1: For n_c of any order and $\gamma < \gamma_o$, there is no solution to the mixed H_2/H_∞ problem.

Proof: See [Rid91a,102], Thm 4.2.2.

Theorem 3.4.2: Assume $n_c \geq n$, and γ is selected such that $\gamma \geq \gamma_2$. Then

i) $K_{mix} = K_{2opt}$

ii) $\alpha^* = \alpha_o$

iii) $\gamma^* = \gamma_2$

Proof: See [Rid91a,101], Thm 4.2.1.

Theorem 3.4.3: Assume $n_c = n$, and γ is selected such that $\gamma_o < \gamma < \gamma_2$. Then the solution to the mixed H_2/H_∞ problem lies on the boundary of the ∞-norm constraint. That is, K_{mix} is such that $\gamma^* = \gamma$.

Proof: See [Rid91a,104], Thm 4.2.3.

Theorem 3.4.4: Assume $n_c = n$. A plot of α^* versus γ^* for $\gamma > \gamma_o$ is monotonically decreasing with γ.

Proof: See [Rid91a,123], Thm 5.1.5.

There are also some important theorems for the suboptimal mixed problem. One of the most important insures that the suboptimal problem converges to the optimal solution.

Theorem 3.4.5: Assume $n_c = n$, and $\gamma > \gamma_o$. Then J_μ given by (3.14) converges to the optimal mixed H_2/H_∞ problem as $\mu \rightarrow 0$.

Proof: See [Rid91a,120], Thm 5.1.2.

One the best ways to visualize all of the above theorems is by examining a typical plot of $\| T_{ed} \|_\infty$ versus $\| T_{zw} \|_2$.

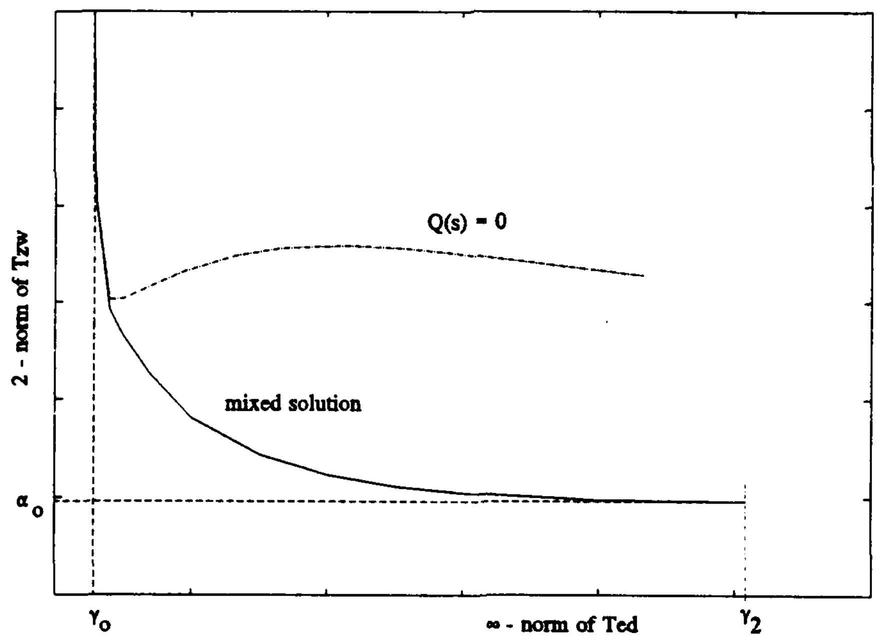

Figure 3-3. Example Full Order Mixed Plot

This plot is taken from actual data from Ridgely's full order SISO example [Rid91a,167]. Several characteristics are immediately evident. All solutions lie inside of (or on) the α_o and γ_o boundaries. The 2-norm gets very large near γ_o. The mixed curve is monotonically decreasing and does not reach α_o until γ_2. The mixed curve literally stops at the (α_o,γ_2) point. Also shown on the plot are the central H_∞ solutions (Q = 0). Notice the large improvement of the mixed solutions over the Q = 0 solutions. This plot is very encouraging from a designer's standpoint. It shows that a significant reduction in the ∞-norm can be achieved by sacrificing only a small amount of H_2 optimality. Physically, this means that there are regions where the system can be made much more robust without giving up very much H_2 performance.

Recall that all the above results assume a full order compensator. What would the mixed plot look like for higher order controllers? Do the higher order controllers exhibit the same characteristics? Is it possible to increase the compensator order enough to achieve α_o at level of γ below γ_2? These are issues that are addressed in the next two chapters.

IV. Increased Order Compensators in Mixed H_2/H_∞ Optimization

4.1 Optimal Mixed Problem

In Chapter III the general mixed H_2/H_∞ problem was developed. During the development, the form of the compensator was assumed, but its order was not specified. Then, after finding the necessary conditions for a minimum, there came a point where the order of the compensator had to be fixed before the analysis could continue. Results for full order compensators have been demonstrated; however, increased order controllers have not yet been thoroughly addressed. Due to the dynamic nature of solutions under the assumption of state availability, it was informally conjectured by Rotea & Khargonekar that compensators of order greater than the plant would be required in the output feedback case [RK91,310] (this conjecture will be shown to be true for certain choices of γ). The issue of higher order compensators is, therefore, well motivated. This chapter extends some of the full order results to compensators of order greater than the plant for the output feedback case, and takes some preliminary steps toward answering the question of optimal compensator order for the mixed H_2/H_∞ problem.

First, when examining the mixed problem, it will be helpful to define three regions based on the chosen level of γ:

Region 1: $0 < \gamma < \gamma_o$

Region 2: $\gamma_o < \gamma < \gamma_2$

Region 3: $\gamma_2 \leq \gamma \leq \infty$

As seen already in the full order case, the nature of the solution is highly dependent on γ. Therefore, each region will be addressed separately.

4.1.1 Region 1. Region 1 can be dismissed immediately, as there is no controller of any order that will meet the ∞-norm constraint since the set of all H_∞ suboptimal controllers is then empty (see also Theorem 3.4.1).

4.1.2 Region 2. Region 2 ($\gamma_o < \gamma < \gamma_2$) is the region of greatest interest since it is here that there may be competing objectives. Recall that in the state feedback case, if γ is chosen such that a solution exists, the absolute minimum of $\|T_{zw}\|_2$ could be achieved while meeting the $\|T_{ed}\|_\infty$ constraint. Under output feedback, this is not true for γ levels lower than γ_2.

Lemma 4.1.1: Assume $n_c \geq n$, and γ is selected such that $\gamma_o < \gamma < \gamma_2$. Under output feedback, it is not possible for the mixed H_2/H_∞ controller to achieve the absolute minimum 2-norm of T_{zw}. That is, $\alpha^* \neq \alpha_o$.

Proof: Assume $\alpha^* = \alpha_o$. Then, by the parameterization of all H_2-(sub)optimal compensators, Q=0, and the compensator is unique (K_{2opt}). γ^* is identically equal to γ_2. This is a contradiction for $\gamma < \gamma_2$. Therefore, if $\gamma < \gamma_2$, then $\alpha^* \neq \alpha_o$ regardless of compensator order. ■

It has been shown that for full order compensators (that is, $n_c = n$), the solution to the mixed H_2/H_∞ problem lies on the boundary of the ∞-norm constraint. In other words, in Region 2, $\gamma^* = \gamma$. This result is also true for higher order compensators.

<u>Lemma 4.1.2</u>: Assume $n_c \geq n$, and γ is selected such that $\gamma_o < \gamma < \gamma_2$. The solution to the mixed H_2/H_∞ problem lies on the boundary of the ∞-norm constraint ($\gamma^* = \gamma$).

Proof: For $n_c = n$, see Theorem 3.4.3.

For $n_c > n$, begin by assuming that: a) the solution is off the boundary (Y=0), and that b) K(s) is a minimal realization. With the assumption that the solution is off the boundary, the first order necessary conditions (with the sub-blocks expanded) are given by:

$$X_{12}^T Q_{12} + X_2 Q_2 = 0 \tag{4.1}$$

$$X_{12}^T Q_1 C_y^T + X_2 Q_{12}^T C_y^T + X_{12}^T V_{12} + X_2 B_c V_2 = 0 \tag{4.2}$$

$$B_u^T X_1 Q_{12} + B_u^T X_{12} Q_2 + R_{12}^T Q_{12} + R_2 C_c Q_2 = 0 \tag{4.3}$$

$$A Q_1 + Q_1 A^T + B_u C_c Q_{12}^T + Q_{12} C_c^T B_u^T + V_1 = 0 \tag{4.4}$$

$$A Q_{12} + Q_{12} A_c^T + B_u C_c Q_2 + Q_1 C_y^T B_c^T + V_{12} B_c^T = 0 \tag{4.5}$$

$$A_c Q_2 + Q_2 A_c^T + B_c C_y Q_{12} + Q_{12}^T C_y^T B_c^T + B_c V_2 B_c^T = 0 \tag{4.6}$$

$$A^T X_1 + X_1 A + C_y^T B_c^T X_{12}^T + X_{12} B_c C_y + R_1 = 0 \tag{4.7}$$

$$A^T X_{12} + X_{12} A_c + C_y^T B_c^T X_2 + X_1 B_u C_c + R_{12} C_c = 0 \tag{4.8}$$

$$A_c^T X_2 + X_2 A_c + C_c^T B_u^T X_{12} + X_{12}^T B_u C_c + C_c^T R_2 C_c = 0 \tag{4.9}$$

Note that these equations are valid and must be satisfied regardless of the compensator order that is chosen.

Consider equation (4.6). Since $\tilde{Q}_2 \geq 0$, from [KJ72,147-148] it follows that

$$Q_2 \geq 0 \tag{4.10}$$

$$Q_{12} = Q_{12} Q_2^{+} Q_2$$

where Q_2^+ is the Moore-Penrose pseudoinverse of Q_2. Therefore, (4.6) can be rewritten as

$$(A_c+B_cC_yQ_{12}Q_2^+)Q_2 + Q_2(A_c+B_cC_yQ_{12}Q_2^+)^T + B_cD_{yw}D_{yw}^TB_c^T = 0 \qquad (4.11)$$

Now, since (A_c,B_c) was assumed controllable (minimal realization), by ([Won85], Lemma 2.1) it follows that $(A_c+B_cC_yQ_{12}Q_2^+,B_c)$ is also controllable. Further, since D_{yw} has full row rank, $(A_c+B_cC_yQ_{12}Q_2^+,B_cD_{yw})$ is controllable. Now, using the dual of ([Won85], Lemma 12.2), equations (4.10) and (4.11) imply $Q_2>0$. Therefore, Q_2^{-1} exists. A similar argument applied to (4.9) implies that X_2^{-1} exists.

Now, examine equation (4.1), rewritten as

$$X_2Q_2 = -X_{12}^TQ_{12} \qquad (4.12)$$

For the case of $n_c > n$, X_{12} and Q_{12} are non-square, $X_{12} \in \mathbf{R}^{n \times n_c}$ and $Q_{12} \in \mathbf{R}^{n \times n_c}$. Therefore, the highest rank that the product $X_{12}^TQ_{12}$ can have is n, and by (4.12) this implies $\operatorname{rank}(X_2Q_2) \leq n$. Therefore, X_2^{-1} ($X_2 \in \mathbf{R}^{n_c \times n_c}$) and/or Q_2^{-1} ($Q_2 \in \mathbf{R}^{n_c \times n_c}$) do not exist. However, it was already shown that both X_2^{-1} and Q_2^{-1} exist; thus, a contradiction. At least one of the assumptions was violated. There are three possibilities, all of which yield the same conclusion:

1) If K(s) is a minimal realization, then $Y \neq 0$. Therefore, the solution lies on the boundary.

2) If $Y \equiv 0$, then K(s) is not a minimal realization. It immediately follows (see [Rid91a,106]) that K(s) is just a state space transformation of the unique n^{th} order compensator K_{2opt}, plus arbitrary pole-zero cancellations. Again, the solution lies on the boundary.

3) If both assumptions were incorrect, then K(s) is not a minimal realization and $Y \neq 0$. The solution still lies on the boundary. ■

With these results in hand, the characteristics of Region 2 can be summarized as follows:

<u>Theorem 4.1.1</u>: Assume $n_c > n$, and γ is selected such that $\gamma_o < \gamma < \gamma_2$. Then,

i) $\alpha_o < \alpha^*_{n_c} \leq \alpha^*_n$

(where α^*_m is α^* for an m^{th} order controller)

ii) $\gamma^* = \gamma$

iii) $n^* \geq n$

Proof: i) $\alpha_o < \alpha^*_{n_c}$ follows immediately from Lemma 4.1.1. If the higher order K_{mix} is simply a non-minimal realization of the full order K_{mix}, $\alpha^*_{n_c} = \alpha^*_n$. However, there is an infinite set of H_2-suboptimal controllers for $\alpha > \alpha_o$, so further reduction of α^* may be possible. This reduction is indeed possible, as will be shown by example. It is conjectured that condition i) can be strengthened to $\alpha^*_{n_c} < \alpha^*_n$ in general, but this still requires formal proof.

Condition ii) was shown in Lemma 4.1.2. The solution lies on the boundary of the ∞-norm constraint for increased order controllers.

Condition iii) follows from i). However, to prove it more formally, assume the optimal order of K_{mix} is n. Then, there can be no other compensator of any order that can achieve a value of $\|T_{zw}\|_2$ lower than α^*_n. However, as will be shown in the examples, higher order compensators can be used to achieve a lower α^* than that achieved with an n^{th} order controller. This contradicts the assumption that the optimal order must be n. Therefore, in general, the optimal order may be greater than the order of the plant. In fact, if the conjecture that $\alpha^*_{n_c} < \alpha^*_n$ is shown to be true, iii) can immediately be strengthened to $n^* > n$. ■

<u>4.1.3 Region 3</u>. For Region 3 ($\gamma_2 \leq \gamma \leq \infty$), the ∞-norm constraint is inactive and the problem reduces to an uncons[illegible]d H_2 optimization problem for which optimal order is known.

<u>Theorem 4.1.2</u>: Assume $n_c \geq n$ and γ is selected such that $\gamma \geq \gamma_2$. Then

i) $K_{mix} = K_{2opt}$

ii) $\alpha^* = \alpha_o$

iii) $\gamma^* = \gamma_2$

iv) $n^* = n$

Proof: For $n_c = n$, the H_2 optimal controller is unique and is given by the results in Section 2.1 with $Q = 0$. Using this compensator, by definition $\|T_{zw}\|_2 = \alpha_o$ and $\|T_{ed}\|_\infty = \gamma_2$. Since α_o is the absolute minimum achievable 2-norm of T_{zw}, and since the constraint $\|T_{ed}\|_\infty \leq \gamma$ is trivially satisfied when $\gamma \geq \gamma_2$, condition i) is true for $n_c = n$.

For $n_c > n$, K_{2opt} makes $\|T_{zw}\|_2 = \alpha_o$, which is the minimum achievable value of $\|T_{zw}\|_2$ using any compensator of any order. Thus it must be the solution and condition i) is true for $n_c > n$.

Conditions ii) and iii) follow immediately from the definitions given in Section 3.1.

Condition iv) holds because K_{2opt} is a unique compensator of order n. If n_c is chosen such that $n_c > n$, K_{mix} must be nothing more than a state space transformation of the unique compensator plus arbitrary pole-zero cancellations. ■

Some extensions from the full order case to the increased order case have now been made. However, while the optimal order of the mixed solution is not, in general, the order of the plant, the ultimate question of what the optimal order is remains to be proven.

4.2 Suboptimal Mixed Problem

As described in Section 3.3, due to difficulties with the numerics of the necessary conditions for the optimal mixed problem, a suboptimal approach is used when solving the problem numerically. However, since this is different problem, the nature of the suboptimal problem needs to be addressed. A main concern is: as $\mu \rightarrow 0$, does the suboptimal problem converge to the optimal? Simply because the $\mu = 0$ case reduces to the optimal mixed problem does not imply that the function J_μ (Equation (3.14)) converges to the function J (Equation (3.4)) as μ approaches zero. It is possible for a function to approach a certain value in the limit and have a discontinuity at that point (that is, the value of the function at the point is completely different than the value of the limit). For full order compensators, it has been shown that the suboptimal problem does approach the optimal as $\mu \rightarrow 0$ (see Theorem 3.4.5). However, before numerical solutions are obtained for higher order compensators, this same convergence needs to be demonstrated for $n_c > n$.

Lemma 4.2.1: Let $G(s) = C(sI-A)^{-1}B$ with A stable. Define $\|G(s)\|_\infty \equiv \gamma_n$ and let $\gamma \geq \gamma_n$. Define $\varepsilon = \gamma - \gamma_n \geq 0$. Then as $\varepsilon \rightarrow 0$, the value of the entropy, $I[G(s),\gamma]$ (which is singular at $\varepsilon = 0$), converges to

$$tr[X_n C^T C]$$

where X_n is the neutrally stabilizing solution to the ARE

$$AX_n + X_n A^T + \gamma_n^{-2} X_n C^T C X_n + BB^T = 0$$

Proof: Immediate, from the fact that Theorem 5.1.1 and its proof, given in [Rid91a,118-119], are completely general. Even though the assumption that $n_c = n$ was made prior to this theorem, the theorem as stated is not a function of compensator order. ■

This result is very important for the suboptimal mixed problem. It is well known that the entropy at $\varepsilon = 0$ is infinite. However, until this theorem was given, it was generally accepted in the literature that the entropy continuously approaches infinity in the limit as $\varepsilon \to 0$. If this was true, then the added term $\mathrm{tr}[Q_\infty \tilde{C}_e^T \tilde{C}_e]$ in the suboptimal mixed H_2/H_∞ cost function would become infinite as μ approaches zero, because the neutrally stabilizing solution to the related Riccati equation (3.12) is required in the optimal mixed solution. However, since the entropy approaches a finite value in the limit, it is possible for the suboptimal mixed problem to converge.

Theorem 4.2.1: Assume $n_c \geq n$, and $\gamma > \gamma_0$. Then J_μ given by (3.14) converges to the optimal mixed H_2/H_∞ problem as $\mu \to 0$.

Proof: For $n_c = n$, see Theorem 3.4.5.

For $n_c > n$, proof is immediate upon recognition that Theorem 5.1.2 and its proof given in [Rid91a,120] are completely general and do not depend on compensator order. The transfer function discussed in the proof is the closed-loop transfer function T_{ed}, and no restrictions are placed on its order. For convenience, a rough outline of the proof is as follows:

- T_{ed} is stable and strictly proper at all values of μ
- it follows that the second term in (3.14) is $I[T_{ed},\gamma]$ for $\mu \neq 0$
- from Lemma 4.2.1 the entropy converges to a finite bound as $\mu \rightarrow 0$
- the term $\mu\ \mathrm{tr}[Q_\infty \tilde{C}_e^T \tilde{C}_e]$ therefore approaches zero as $\mu \rightarrow 0$
- thus, $J_\mu \rightarrow J$ as $\mu \rightarrow 0$ ■

Even though no rigorous proofs were required to extend the full order case to the higher order case in proving the convergence of the suboptimal mixed problem, this was an important issue that needed to be shown. The suboptimal mixed problem does indeed converge to the optimal for higher order compensators. Having shown some key theoretical results, now consider the numerical solution of the mixed H_2/H_∞ problem.

V. Numerical Solution

5.1 Davidon-Fletcher-Powell Algorithm

Since a closed-form solution of the mixed H_2/H_∞ problem is not available, the problem must be solved numerically. There are many numerical optimization algorithms available. Ideally, a second order gradient method (Newton's procedure) might be desired, but by definition, this type of algorithm requires the second partials of the function being minimized. For the mixed H_2/H_∞ problem, these second derivatives are fourth-order tensors. Therefore, due to memory limitations and excessive execution times (as discovered by [Rid91a]), this approach was avoided. Instead, an algorithm developed by Davidon and further described and refined by Fletcher and Powell was used (see [Fox71,104-109]). The Davidon-Fletcher-Powell (DFP) algorithm is a very powerful quadratically convergent first-order method. It does not require the second derivatives, but it does calculate estimates of these derivatives. These estimates are used to form a variable metric which is improved with each iteration. The flow diagram shown in Figure 5-1 outlines the basic progression of the algorithm. **X** is a column vector containing the unknown variables. For the mixed H_2/H_∞ problem, these are the A_c, B_c, and C_c matrices stretched into a vector. F(**X**) is the function being minimized. For the mixed problem, $F \equiv J_\mu$. The gradient,

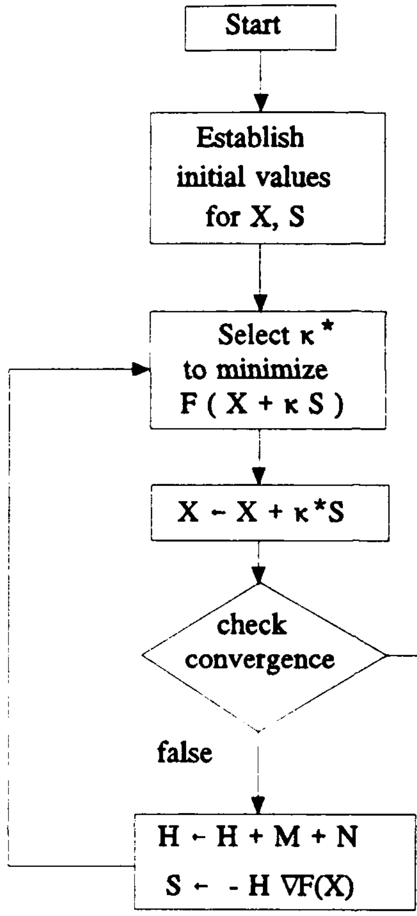

Figure 5-1. DFP Flow Diagram

∇F(**X**), for the mixed problem are the partial derivatives of the Lagrangian $\mathscr{L}_\mu$ with respect to A_c, B_c, and C_c stretched into a column vector. **S** is the matrix that specifies the direction that the current guess of **X** needs to move in order to reduce F(**X**). Typically, in a true second-order method, $\mathbf{S} = -\mathbf{J}^{-1}\,\nabla F(\mathbf{X})$, where **J** is the Hessian (second derivative matrix). As already mentioned, DFP estimates $\mathbf{J}^{-1}$. This estimate is a symmetric, positive definite matrix defined as **H**. Thus, in DFP, $\mathbf{S} = -\mathbf{H}\,\nabla F(\mathbf{X})$. κ is the metric that controls the size of the step tc be taken in the **S** direction. **M** and **N** are matrices that are used to calculate **H**, and will be defined shortly.

The iteration proceeds as follows:

1) Begin with an initial **X** vector and **H**. The initial **X** that was used was the initial guess of the compensator. Section 5.2 discusses how this initial guess was determined. The initial **H** was chosen to be the identity matrix; thus the initial direction of movement was simply the negative of the gradient at that point.

2) Compute $\mathbf{X}_{q+1} = \mathbf{X}_q + \kappa^*_q \mathbf{S}_q$, where κ^*_q minimizes $F(\mathbf{X}_q + \kappa_q \mathbf{S}_q)$. The method used for finding κ^* was a simple one-dimensional search involving a doubling/bisection technique. For an outline of this technique see [Rid91,130]. More sophisticated methods were attempted in order to speed up execution times (this is where most of the computations are required); however, this method proved to be the most reliable in the face of such a difficult function.

3) If the algorithm has not converged, compute $\mathbf{H}_{q+1} = \mathbf{H}_q + \mathbf{M}_q + \mathbf{N}_q$, where

$$Y_q \equiv \nabla F(X_{q+1}) - \nabla F(X_q)$$

$$M_q = \kappa_q^* \frac{S_q S_q^T}{S_q^T Y_q}$$

$$N_q = -\frac{(H_q Y_q)(H_q Y_q)^T}{Y_q^T H_q Y_q}$$

4) Finally, calculate the new S by $\mathbf{S}_{q+1} = - \mathbf{H}_{q+1} \nabla F(\mathbf{X}_{q+1})$ and return to the κ^* calculation step with the new **X** and **S**.

Convergence of the solution is declared when

$$\frac{\nabla F_q^T H_q \nabla F_q}{|F(X_q)|} < \epsilon$$

[Fox71,104-109]

where ϵ is a small positive number that is defined by the user prior to execution (values of 10^{-6} were typical).

This algorithm was coded into FORTRAN to run on a VAX mainframe. The FORTRAN program is given in the appendix. Note that in the code, the variable κ is called α. This is because the program was written to be consistent with the nomenclature given in [Fox71]. However, in order to avoid confusion with the definitions of α given throughout this work, the discussion in this section uses the nomenclature κ. A PRO-MATLAB™ version of DFP was used by Ridgely in his work. The FORTRAN version was found to be about 7-10 times faster.

An outline of the process for using the DFP program to solve the mixed H_2/H_∞ problem is as follows:

1) Select the desired compensator order and level of γ and determine an initial guess for the mixed compensator. As will be shown in the following section, this was not a trivial task. The program is extremely sensitive to the starting point.

2) Start with a relatively high value for μ (usually between 0.1 and 0.5) and run the program until it converges.

3) Use the compensator found in step 2) as a new initial guess and re-start the program at a lower value of μ.

4) Continue the process of reducing μ until the solution converges acceptably close to the optimal mixed solution (values of around 10^{-4} were typically attained for μ). Three items were checked when determining if the solution was "close enough." First, the 2-norm of T_{zw} would be checked to see if it was still becoming smaller. Generally, when convergence was declared, reductions could only be seen in the fifth or six decimal place. Second, since it was known that the solution has to lie on the boundary of the ∞-norm constraint, the ∞-norm of T_{ed} was checked. At "convergence", γ^* was typically within a 10^{-6} tolerance of γ (from below). Finally, the derivatives with respect to A_c, B_c, and C_c were checked to insure they were close to zero (they should be exactly zero for the true solution). Usually, at convergence, most of the elements in these derivative matrices were around 10^{-3} to 10^{-5}.

5.2 Determining an Initial Guess for DFP

As with any numerical technique, the DFP algorithm requires an initial guess of the solution. As discussed in Chapter 3, the central H_∞ (minimum entropy) controller is the solution of the suboptimal mixed problem for $\mu = 1$. This is a full order compensator. Therefore, if the desired order of the mixed solution is the order of the plant, this is an acceptable initial guess for the given level of γ or higher (note that if a central H_∞ compensator designed for a certain level of γ is used as a starting point for a higher γ, it may be too far away from the solution for the numerics to converge even though it is an "acceptable" starting point). While the full order minimum entropy controller is a good initial guess for a full order mixed problem, it cannot be used directly for an increased order problem because its size is incompatible. Therefore, finding a good initial guess for the increased order problem is not quite as straightforward as the full order case.

The method for determining an initial higher order guess in this research utilized the J-Q parameterization of K(s), depicted in Figure 5-2.

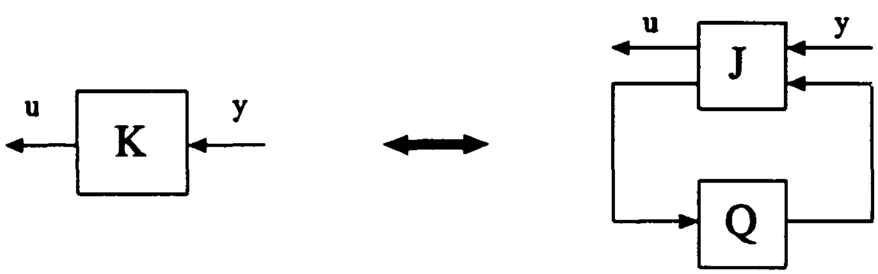

Figure 5-2. Parameterization of K(s)

If K(s) is an H_∞ suboptimal compensator, J is completely known. The central controller (Q = 0) is not the only acceptable choice for a starting guess. As long as Q is stable, strictly proper, and has $\|Q\|_\infty \leq \gamma$, it will produce a compensator that admits a solution to the X, Y, $\tilde{Q}_2$, and Q_∞ Lyapunov and

Riccati equations in the necessary conditions. Therefore, the initial method used was to simply choose a Q with enough arbitrary left-hand plane poles (and an appropriate gain to ensure the ∞-norm bound was met) to give the desired order of K. This worked for the n+1 order compensator in the SISO example, but did not work for any higher orders. Even though the compensator was theoretically acceptable, it was not close enough to the solution for the algorithm to "start moving". It was found that the algorithm is very sensitive to the starting guess.

The method that ended up being the most reliable was less arbitrary than the simplistic approach just discussed. It is possible to take a given compensator K and a J (which is completely specified for a given level of γ) to calculate a Q that, when placed in a feedback loop around J, will produce the given K (see [Rid91a,152-154]. A PRO-MATLAB™ routine (named *K2Q*) that accomplishes this was utilized. It was found that the Q that was calculated always had a relatively high order (it was almost never a minimal realization).

The first step in determining a higher order initial guess was to find the minimum entropy controller for the given γ. If this compensator is input into *K2Q*, the resulting Q would be Q = 0 (as it should be). However, if the central H_∞ compensator was modified slightly (by simply truncating the elements in its state space C matrix after the second or third decimal place) the resulting Q would be non-zero and relatively high order. Then, using a balanced Schur model order reduction, Q was reduced to the amount of states (n_c - n) required to achieve the desired order of K. Thus, the resulting K was essentially a nonminimal realization of the central H_∞ compensator. This process does not guarantee that the $\|Q\|_\infty \leq \gamma$ requirement will be met. Therefore, the state space C matrix for Q would be arbitrarily multiplied by 0.6 in order to reduce Q's maximum singular value. This was always done, even when the ∞-norm bound on Q was already satisfied, because it tended to help the algorithm start moving.

Once a mixed solution was found, it could also be used as a starting guess for a higher compensator order, rather than the minimum entropy controller. Q was always non-zero for the mixed solution, and again it was typically high order. The same process of reducing Q to the desired order and multiplying its C matrix by 0.6 was also used for coming up with the next higher order starting point.

Another variable that has not been discussed yet is μ. This could also be varied in order to help get the program moving. Usually, the first initial guess could be started with $\mu = 0.1$; however, sometimes μ had to be increased to as much as 0.7 in order to start the process. In summary, finding a good initial guess for the higher order case is not a trivial task. Many times, several different attempts had to be made before an acceptable starting point could be found.

5.3 SISO Mixed Optimization Example

Consider the mixed H_2/H_∞ optimization system block diagram that was discussed and developed in Chapter 3 and shown in Figure 5-3.

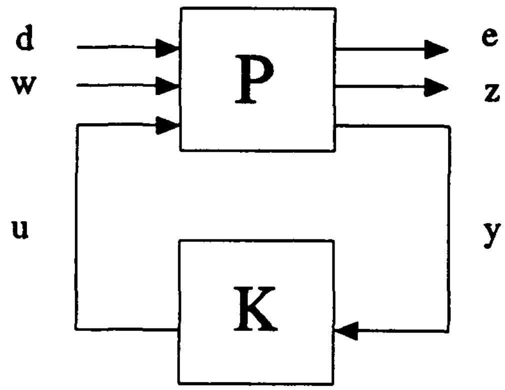

Figure 5-3. Mixed H_2/H_∞ Optimization Block Diagram

For this example, all signals (d, w, e, z, u and y) are assumed to be scalars. The plant P has the state space realization

$$P(s) = \left[\begin{array}{c|ccc} A & B_d & B_w & B_u \\ \hline C_e & D_{ed} & D_{ew} & D_{eu} \\ C_z & D_{zd} & D_{zw} & D_{zu} \\ C_y & D_{yd} & D_{yw} & D_{yu} \end{array}\right]$$

In order to make direct comparisons with a known full order case, the same SISO system that was defined in [Rid91a,130-131] was chosen for this analysis. The state space matrices of the system are:

$$A = \begin{bmatrix} -0.3908 & -0.4565 & 1.2657 \\ 1.4453 & -1.0491 & -1.2077 \\ -0.1288 & 0.6744 & 1.0324 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.0488 \\ 0.3608 \\ 0.3564 \end{bmatrix} \qquad B_w = \begin{bmatrix} 1.4077 \\ 0.9723 \\ -1.6050 \end{bmatrix} \qquad B_u = \begin{bmatrix} -0.4275 \\ -0.4470 \\ -0.9172 \end{bmatrix}$$

$$C_e = [\ 0.9420 \quad 0.0144 \quad 0.1187\ ]$$

$$C_z = [\ -0.0450 \quad 0.3606 \quad 1.8972\ ]$$

$$C_y = [\ -1.5567 \quad -1.9432 \quad -0.0914\ ]$$

$$D_{ed} = [\ 0\ ] \qquad D_{ew} = [\ 0\ ] \qquad D_{eu} = [\ 1.3575\ ]$$

$$D_{zd} = [\ 0\ ] \qquad D_{zw} = [\ 0\ ] \qquad D_{zu} = [\ 0.5781\ ]$$

$$D_{yd} = [\ 0.5185\ ] \qquad Dyw = [\ 0.3899\ ] \qquad D_{yu} = [\ 0\ ]$$

First of all, note that this system does satisfy all the assumptions given in Section 3.1 This is a third order SISO system whose "unweighted" plant, given by

$P_{yu}(s) = C_y(sI-A)^{-1}B_u + D_{yu}$, is open-loop unstable and minimum phase. The singular value plot of P_{yu} is shown in Figure 5-4.

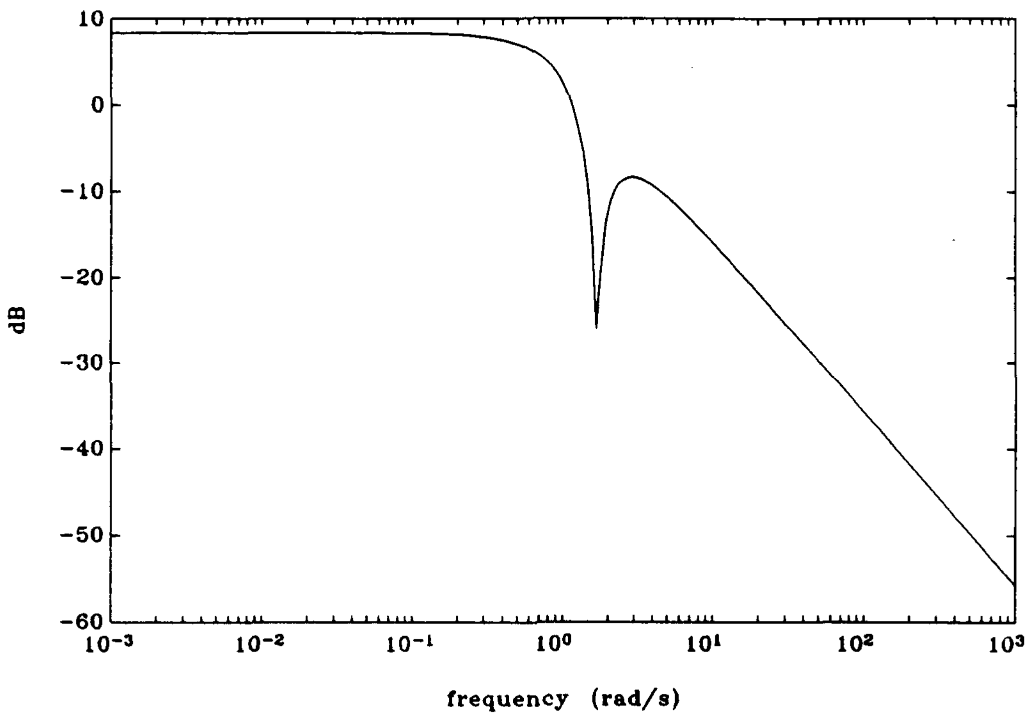

Figure 5-4. Magnitude Plot of SISO Plant

Before beginning the mixed problem, it will be helpful to know the limits of achievable H_2 and H_∞ performance. Consider performing pure unconstrained H_2 and H_∞ optimization on the given plant. Figure 5-5 shows the singular value plot of the unique three-state H_2 optimal controller K_{2opt}. Figures 5-6 and 5-7 are the singular value plots of the corresponding closed-loop transfer functions T_{zw} and T_{ed}. The minimum achievable 2-norm of T_{zw} is $\alpha_o = 9.9263$. The ∞-norm of T_{ed} using K_{2opt} is $\gamma_2 = 4.5364$.

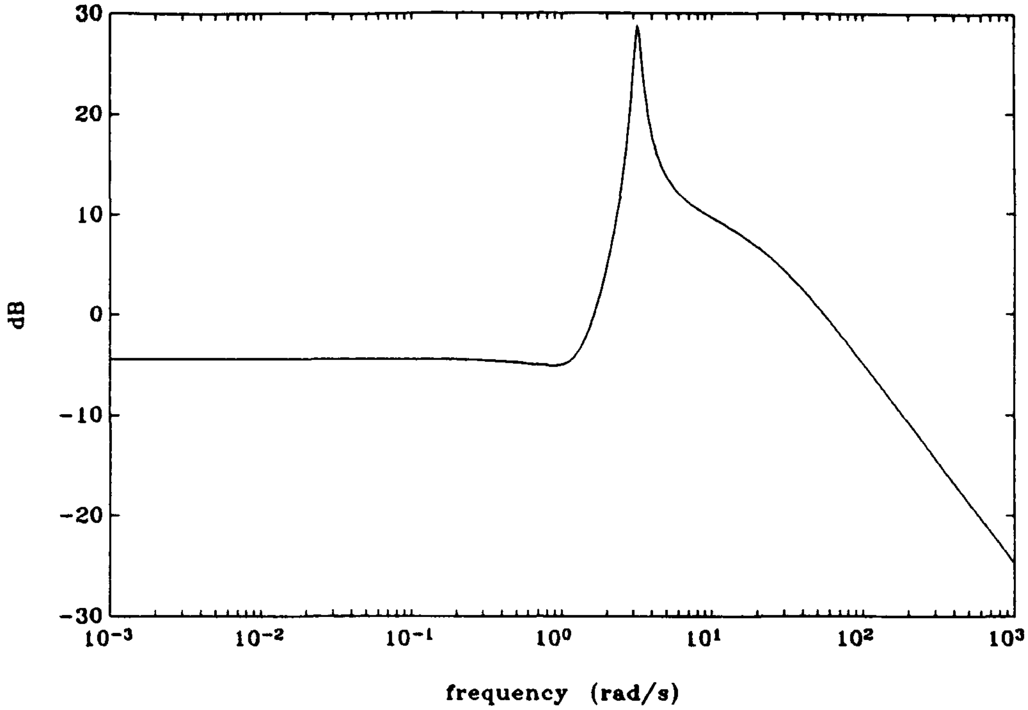

Figure 5-5. Singular Value Plot of K_{2opt}

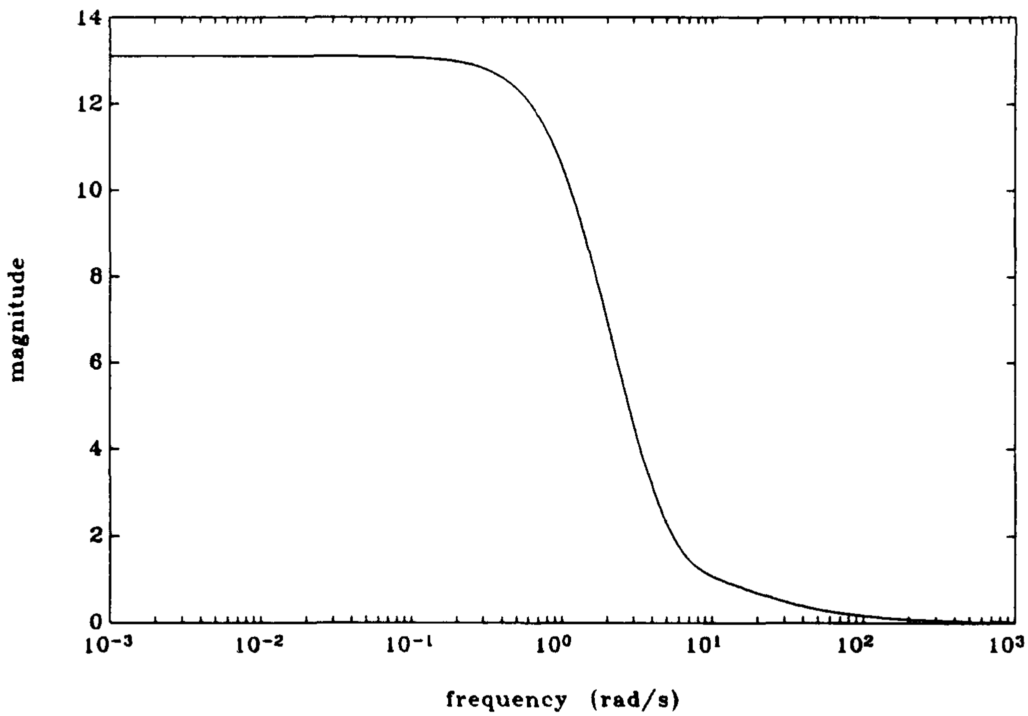

Figure 5-6. Singular Value Plot of T_{zw} for K_{2opt}

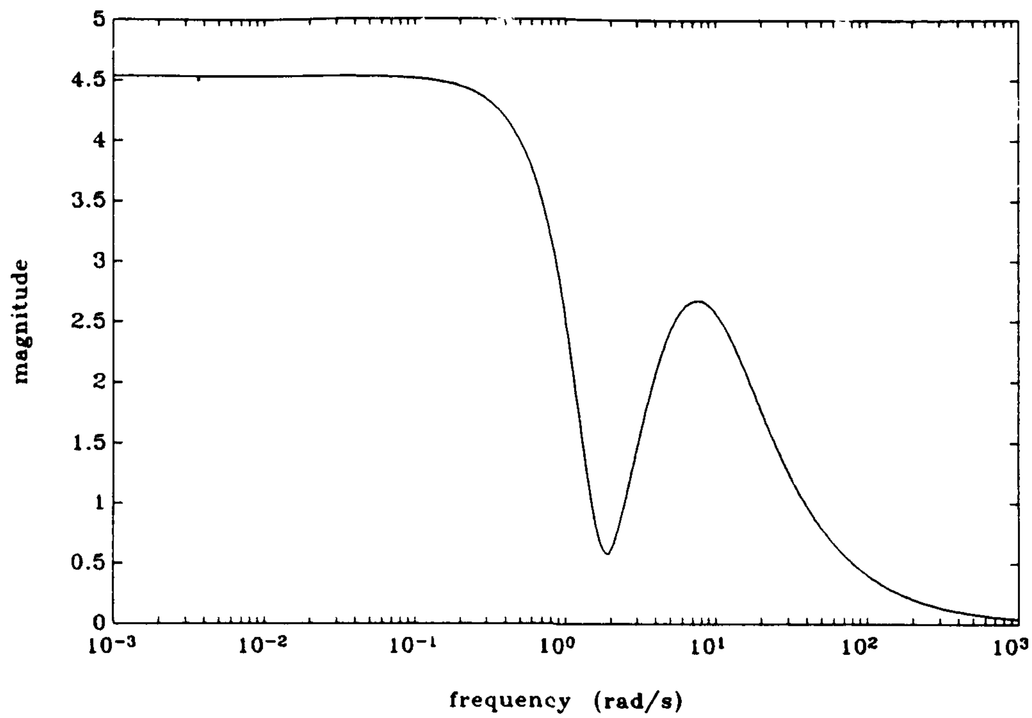

Figure 5-7. Singular Value Plot of T_{ed} for K_{2opt}

Now, if H_∞ optimization is performed, the minimum achievable ∞-norm of T_{ed} is found to be about $\gamma_o \approx 2.1426$. The freedom parameter Q(s) from the parameterization of H_∞ controllers must be specified. The central H_∞ compensator (i.e. minimum entropy controller) is of particular interest since it will be the initial guess for the DFP program (at $\mu = 1$), so choose $Q(s) = 0$. The singular value plot of the H_∞ suboptimal central compensator for $\gamma = 2.1426$ ($K_{\infty 2.1426}$) is given in Figure 5-8. Figures 5-9 and 5-10 are the singular value plots of the corresponding closed-loop transfer functions T_{zw} and T_{ed}. The 2-norm of T_{zw} for $K_{\infty 2.1426}$ is $\|T_{zw}\|_2 = 94.323$, considerably higher than α_o.

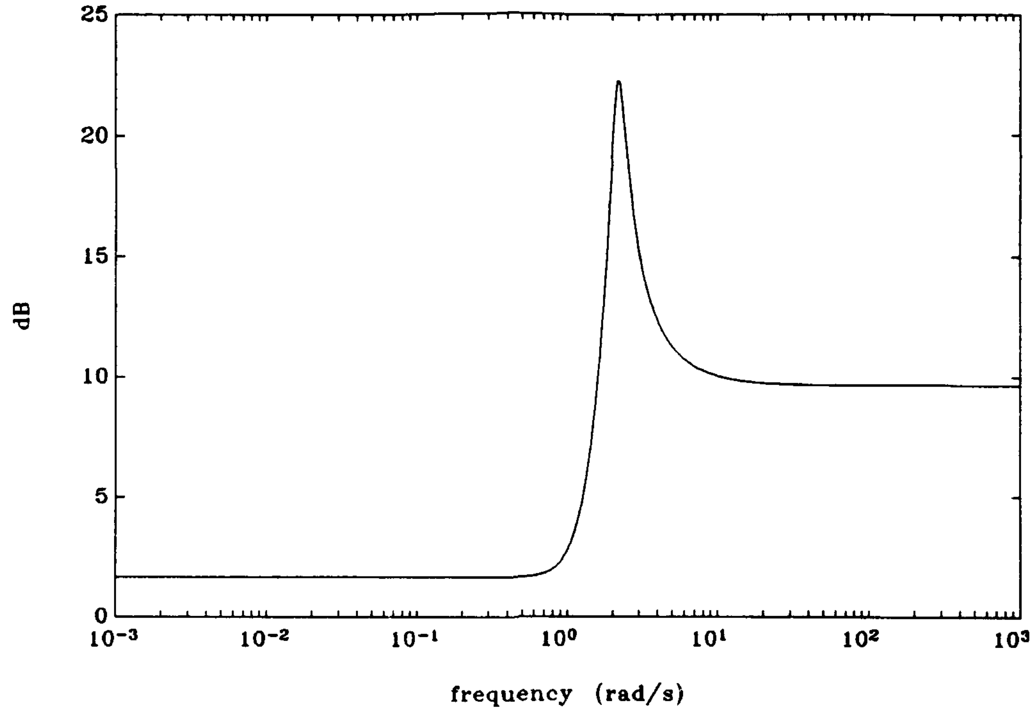

Figure 5-8. Singular Value Plot of $K_{\infty 2.1426}$

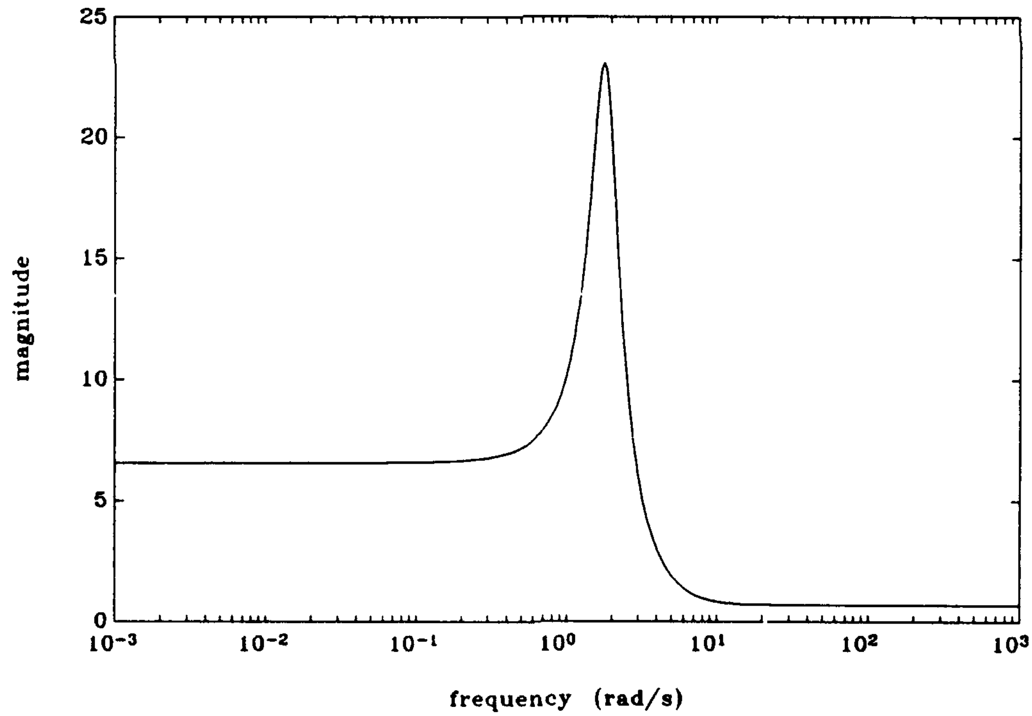

Figure 5-9. Singular Value Plot of T_{zw} for $K_{\infty 2.1426}$

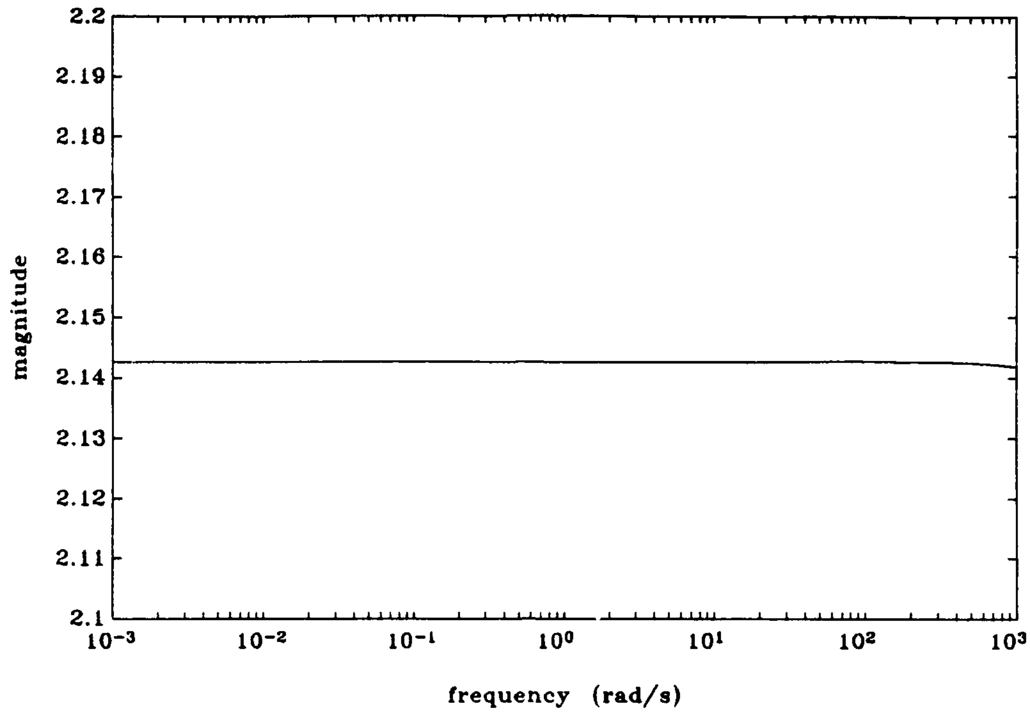

Figure 5-10. Singular Value Plot of T_{ed} for $K_{\infty 2.1426}$

Although it is not evident in these plots, they all have a high frequency roll off. If the optimal H_∞ controller was found, there would be no roll off and $\|T_{zw}\|_2$ would be infinite.

Now consider performing mixed H_2/H_∞ optimization assuming the controller is full order. A brief summary of Ridgely's results are shown in Table 5-1 and Figure 5-11. Table 5-1 shows the values of $\|T_{zw}\|_2$ and $\|T_{ed}\|_\infty$ for the mixed and central H_∞ controllers at varying levels of γ. Notice that $\|T_{ed}\|_\infty = \gamma$ for the mixed controller. Actually, the ∞-norm values are rounded off (typically within 0.00001). Recall that the true mixed solution must lie on the boundary of the ∞-norm constraint. Figure 5-11 shows this data in graphical form.

Table 5-1. SISO Example Full Order Results

γ	(Q = 0) $\Vert T_{ed} \Vert_\infty$	(Q = 0) $\Vert T_{zw} \Vert_2$	(mix) $\Vert T_{ed} \Vert_\infty$	(mix) $\Vert T_{zw} \Vert_2$
2.1426	2.1426	94.323	2.1426	~ 94.323
2.145	2.145	20.387	2.145	~ 20.387
2.15	2.15	16.099	2.15	16.091
2.2	2.2	14.055	2.2	13.868
2.25	2.247	14.076	2.25	13.295
2.35	2.3389	14.305	2.35	12.501
2.5	2.4675	14.622	2.5	11.616
2.75	2.6589	14.953	2.75	10.870
3.0	2.8243	15.107	3.0	10.460
3.25	2.9673	15.161	3.25	10.225
3.5	3.0909	15.164	3.5	10.086
4.0	3.2913	15.107	4.0	9.9539
4.5364	3.4532	15.025	4.5364	9.9263
10.0	3.9948	14.666	4.5364	9.9263
50.0	4.1559	14.557	4.5364	9.9263
100.0	4.1611	14.554	4.5364	9.9263

[Rid91a,166]

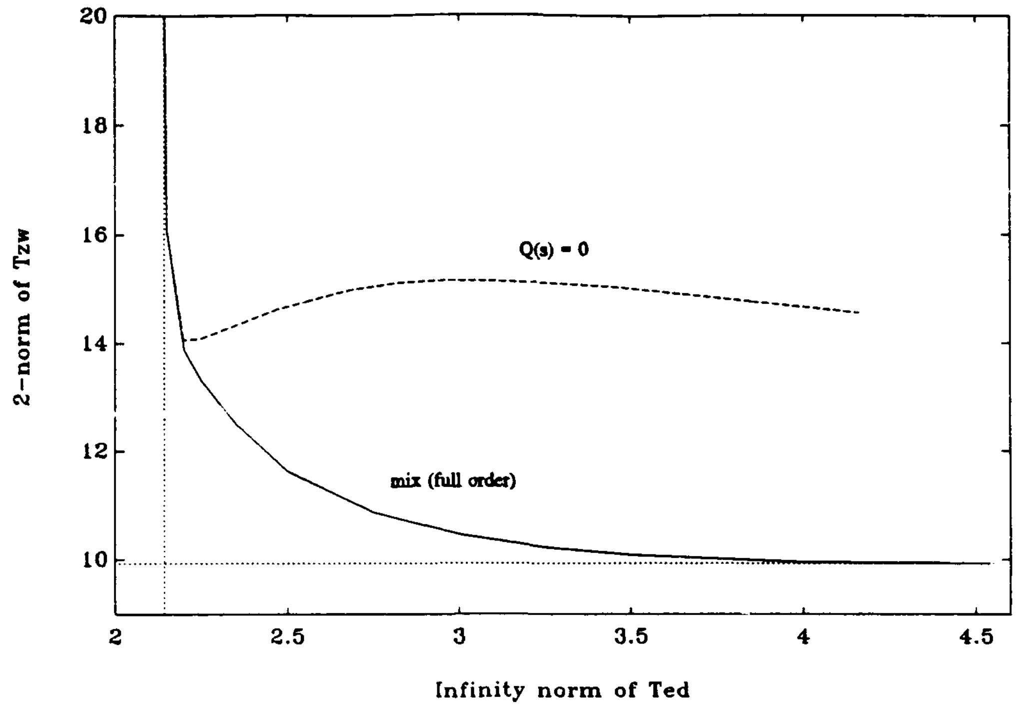

Figure 5-11. SISO Example Full Order and Q(s)=0 Results

[Rid91a,167]

No discussion of these full order results will be given here. Rather, these results are included as a point of reference for the higher order results.

5.3.1 γ=2.5 Results. Now, with all the preliminaries taken care of, consider the case of higher order compensators. Since the nature of the results using a higher order compensator was unknown, the basic approach was to begin with the full order case and continually increase the compensator order until some kind of trend could be recognized. Before this order sweep could be accomplished, the design γ had to be chosen. $\gamma=2.5$ was the first level selected.

Then, compensators of the following orders were obtained by running DFP: 3, 4, 5, 6, 7, 8, 9, 12, 18. Table 5-2 shows a summary of the higher order results for $\gamma = 2.5$. Note that the ∞-norm values are essentially the same as γ. They are actually slightly less than γ (typically by about 10^{-6}).

Table 5-2. Higher Order Results, γ=2.5

Compensator Order	α^*	γ^*
3 (full)	11.61625	2.5
4	11.56329	2.5
5	11.52972	2.5
6	11.51510	2.5
7	11.50500	2.5
8	11.49373	2.5
9	11.48754	2.5
12	11.48654	2.5
18	11.48647	2.5

Figure 5-12 shows this same data on a graph. Note that since only integer order compensators are allowed, this is not a continuous curve. In [Rid91a] it was shown that $\alpha^* = 11.507$ for n_c=9 at γ=2.5 (this was the only higher order result given). The slight discrepancy is easily accounted for by the fact that the question of "close enough" to the true solution is fairly arbitrary. Apparently, the solution given here is converged a little more. In fact, it is possible to very slightly reduce the values given here even more.

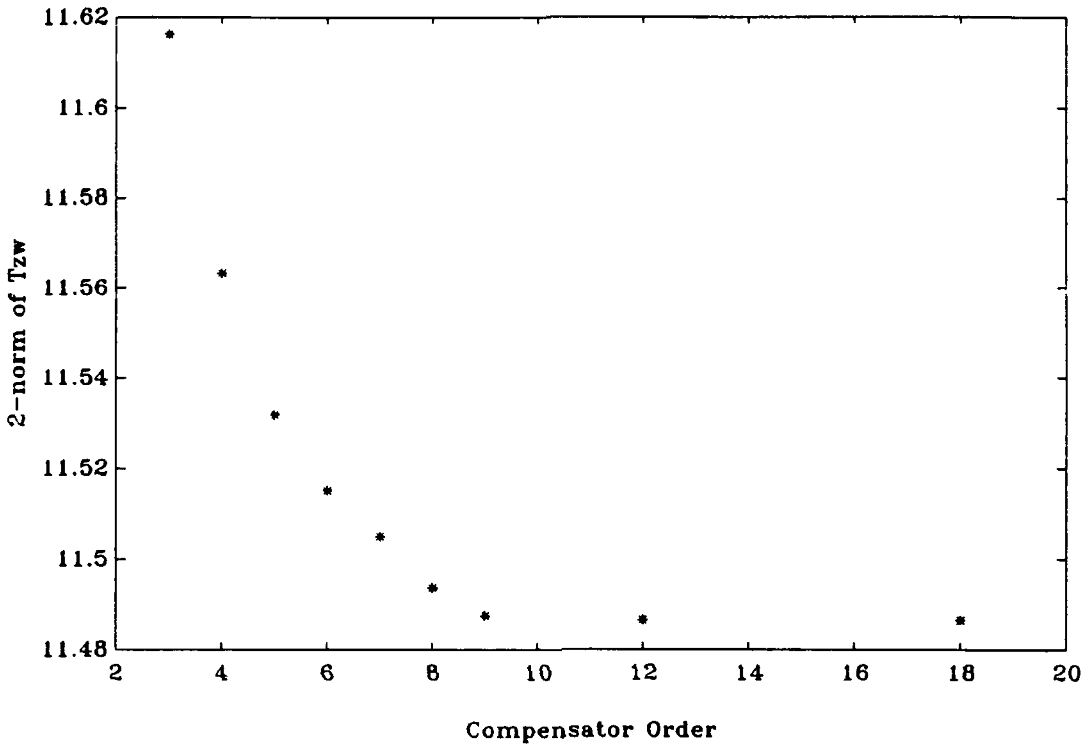

Figure 5-12. Higher Order Results, $\gamma=2.5$

One observation can immediately be made: in general, the optimal order is definitely not the order of the plant. This graph clearly shows that as order is increased beyond full order (n=3), the value of the 2-norm of T_{zw} decreases. There does appear to be an order beyond which meaningful reductions in α^* are no longer attained (i.e. $n_c=9$). Also, in this example, α^* is strictly monotonically decreasing with increasing order. This behavior has not been proven in general (if it was, it would prove that the optimal order is infinite). However, it does appear to be true in this example.

Figures 5-13, 5-14, and 5-15 show the singular value plots of the mixed H_2/H_∞ compensators with increasingly higher orders. Notice first that they all have the same basic shape. The sharp peak at about 2.5 rad/sec decreases in magnitude with increasing order. Also, as might be expected from Figure 5-12, there does not appear to be much difference between the 9, 12, and 18-state compensators (at least in terms of frequency response). Whether or not these compensators are actually different realizations of the same 9th order controller will be discussed later. From these plots, they certainly appear to be essentially the same.

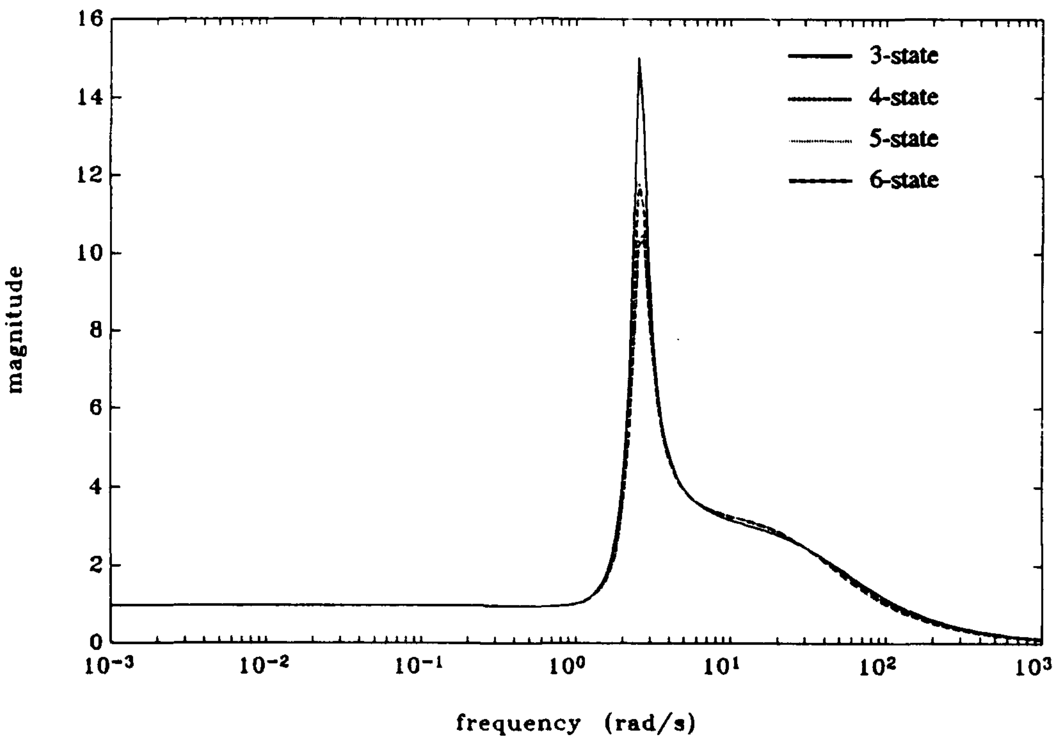

Figure 5-13. Singular Value Plots of K_{mix} (3,4,5,6-state)

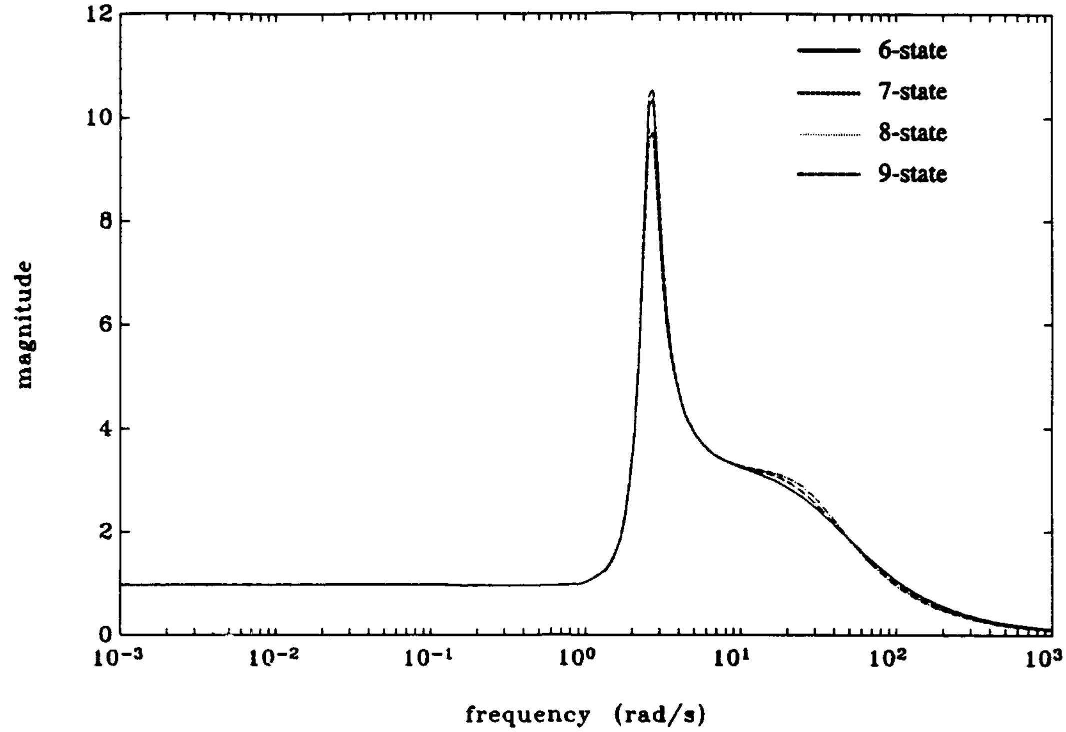

Figure 5-14. Singular Value Plots of K_{mix} (6,7,8,9-state)

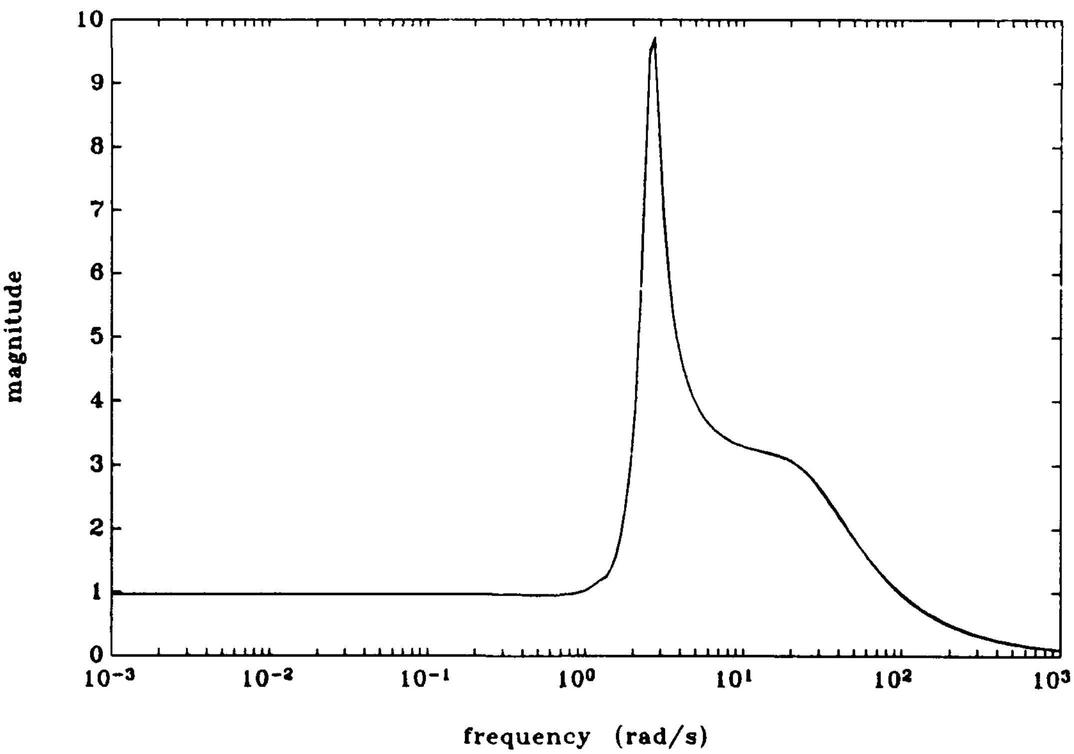

Figure 5-15. Singular Value Plots of K_{mix} (9,12,18-state)

Figures 5-16, 5-17, and 5-18 show the singular value plots of T_{zw} for the higher order mixed solutions. In these plots it is apparent that the higher order compensators decrease the 2-norm of T_{zw} by making the peak in the curve progressively narrower. Even though the peak rises slightly, the net change in area under the curve decreases. Again, there are no distinguishable differences in the 9, 12 and 18-state results. Figures 5-19, 5-20, and 5-21 show the singular value plots of T_{ed} for the higher order mixed solutions. These plots are the most interesting because they best demonstrate the value of the higher order controllers. In [Rid91a], it was shown for the full order case that the mixed solution tries to recover to the H_2 optimal solution. This makes sense since both problems have the same performance objective. However, since the mixed problem has a constraint that must be met, it will try to match the true optimal solution as best as it can while satisfying the constraint. This can be seen dramatically in Figure 5-22. This plot shows the H_2 optimal, the central H_∞, and the full order mixed solutions. Notice how the mixed solution tries to recover to the H_2 solution. In the regions where the H_2 curve exceeds the ∞-norm bound, the mixed solution basically lies right on the ∞-norm boundary. Figure 5-23 shows an expanded view of this same plot with the 9-state solution included (and Q=0 omitted). The extra degrees of freedom provided by the higher order compensator enable a better recovery of the H_2 solution. As can be seen, the 9-state curve makes a much sharper turn where the H_2 and Q=0 curves intersect. It also dips further down into the notch.

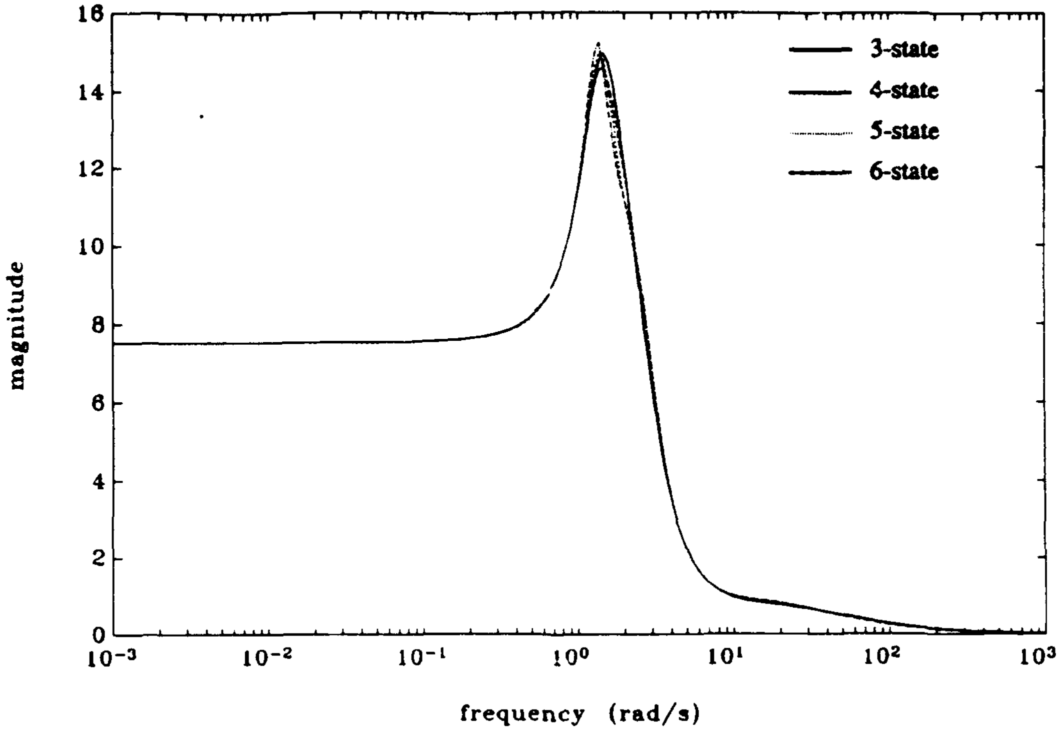

Figure 5-16. Singular Value Plots of T_{zw} for $\gamma=2.5$
(3,4,5,6-state)

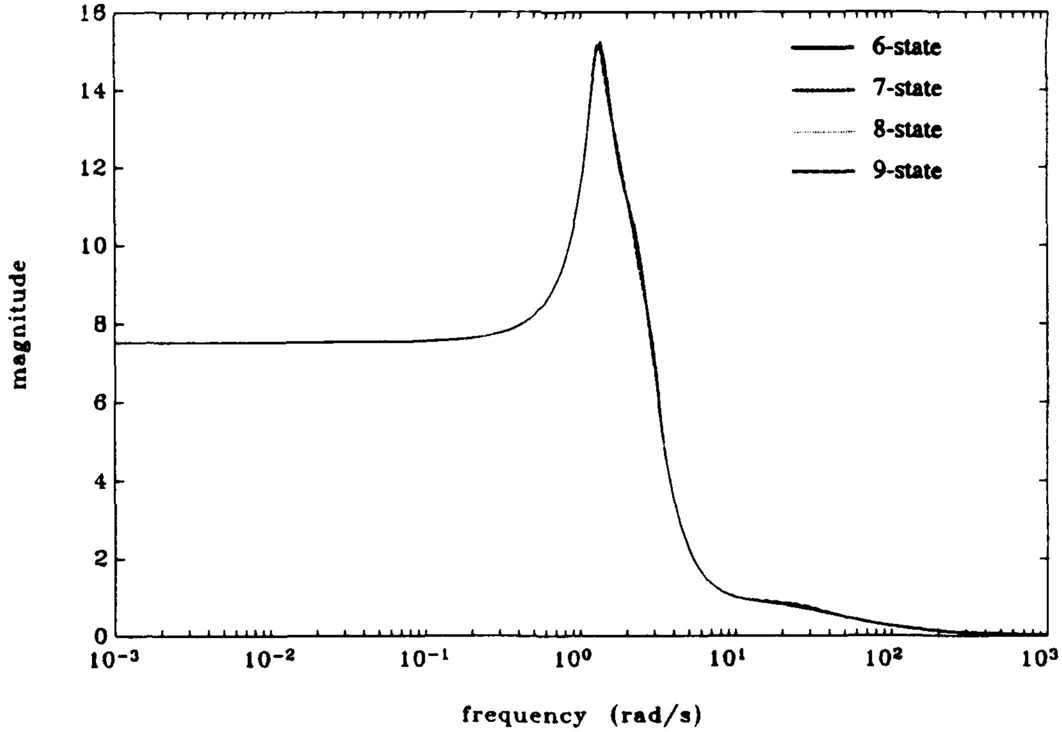

Figure 5-17. Singular Value Plots of T_{zw} for $\gamma=2.5$
(6,7,8,9-state)

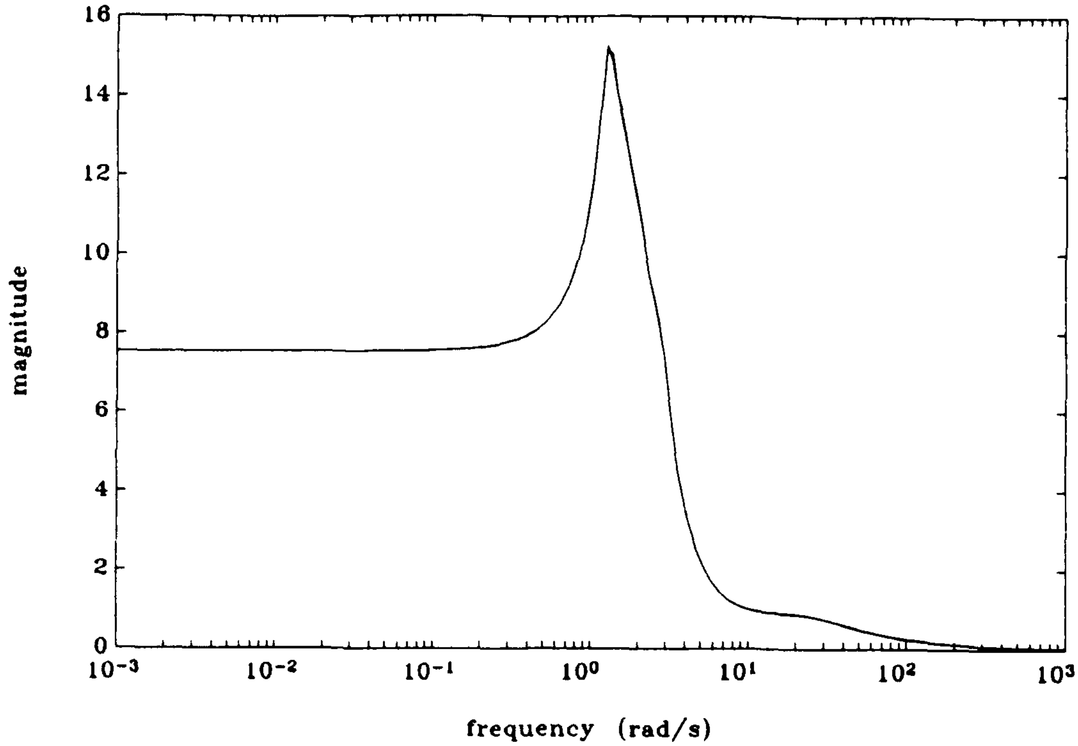

Figure 5-18. Singular Value Plots of T_{zw} for $\gamma=2.5$
(9,12,18-state)

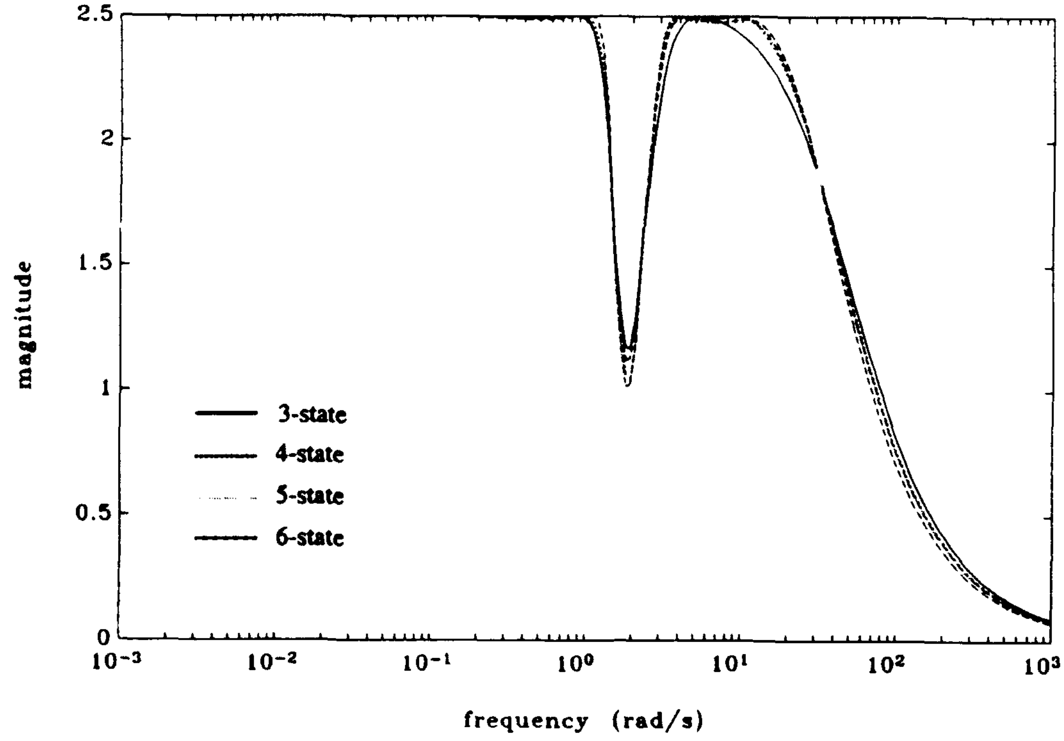

Figure 5-19. Singular Value Plots of T_{ed} for $\gamma=2.5$
(3,4,5,6-state)

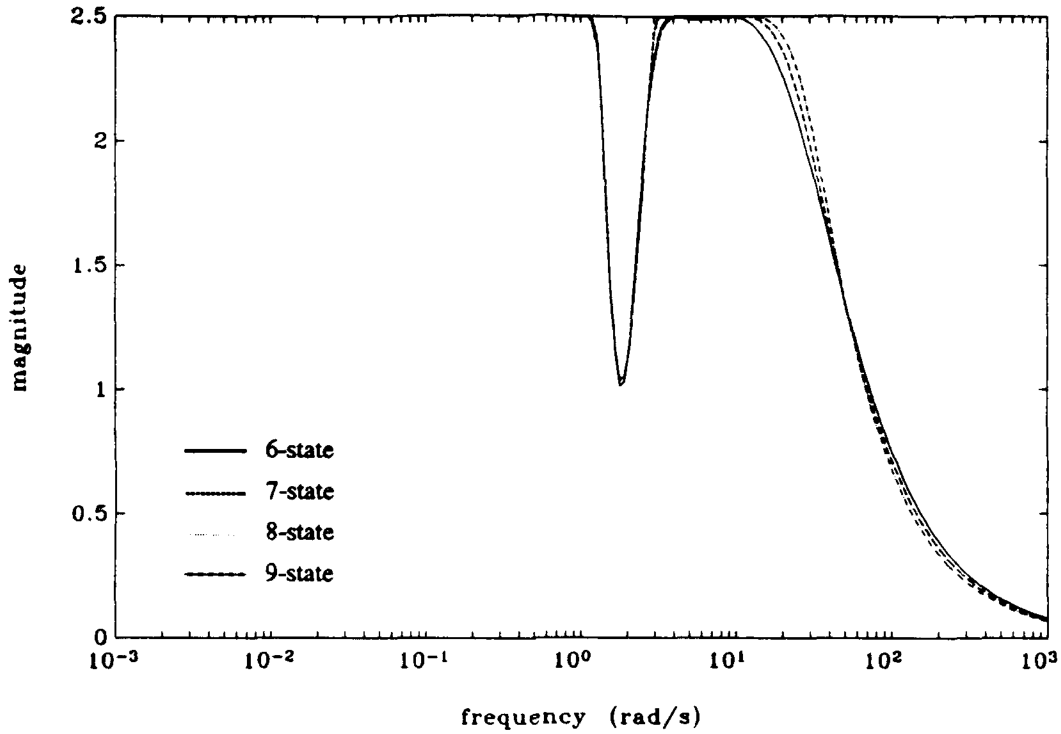

Figure 5-20. Singular Value Plots of T_{ed} for $\gamma=2.5$
(6,7,8,9-state)

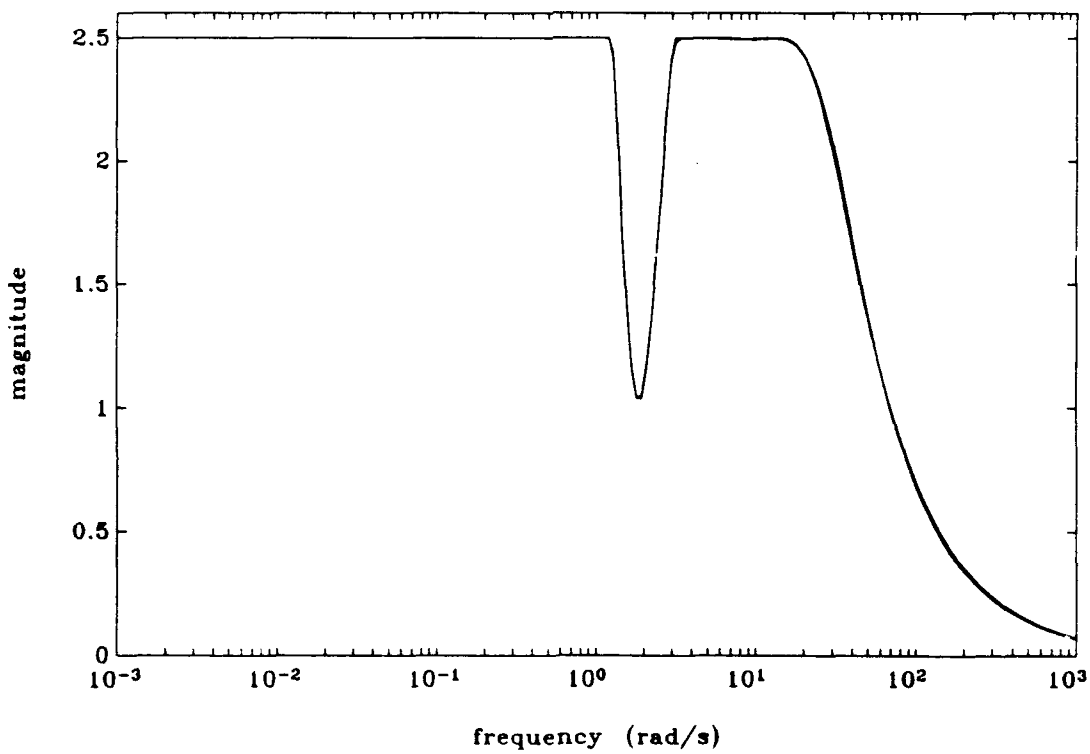

Figure 5-21. Singular Value Plots of T_{ed} for $\gamma=2.5$
(9,12,18-state)

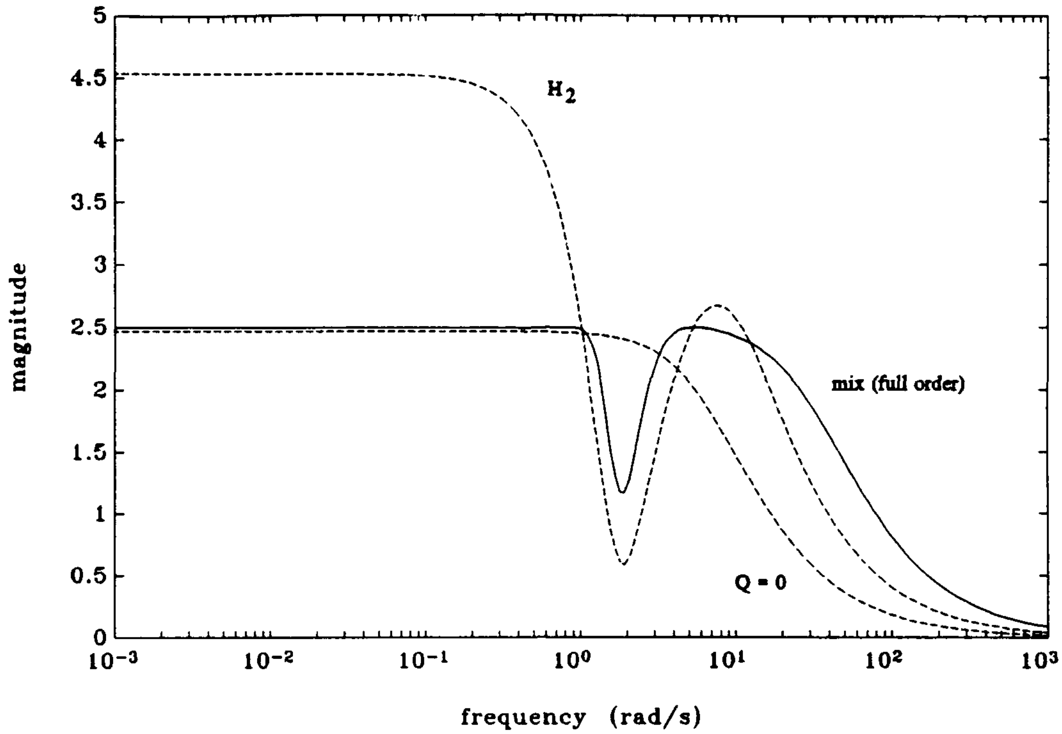

Figure 5-22. T_{ed} Comparison Plot for $\gamma=2.5$

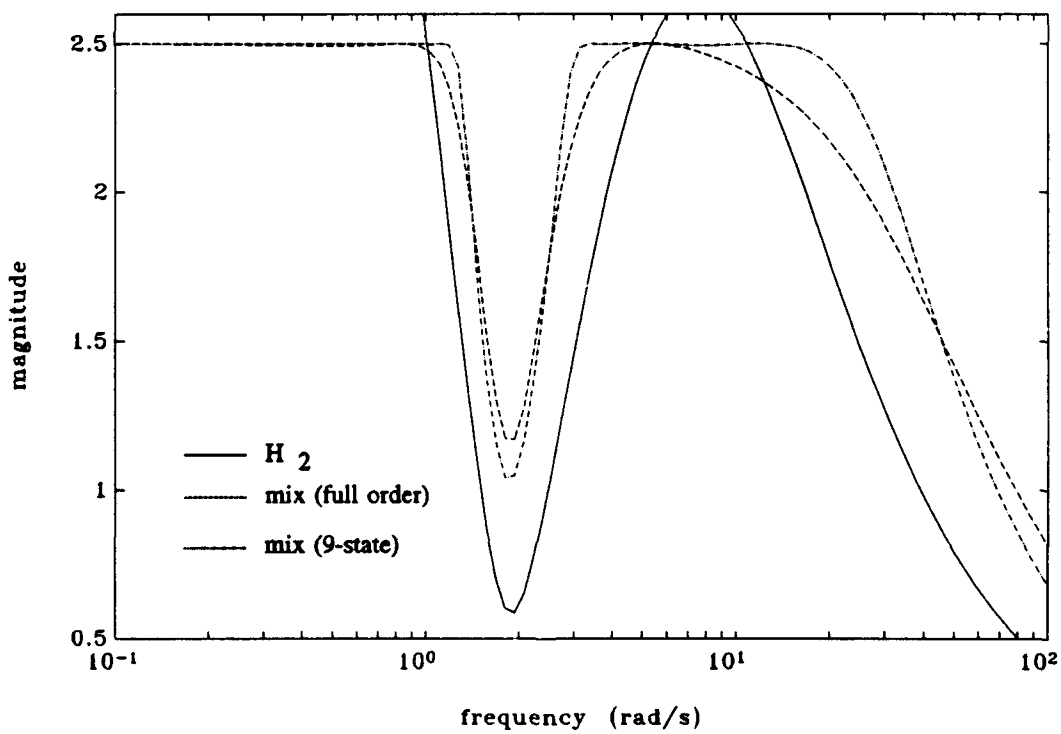

Figure 5-23. T_{ed} Comparison Plot Expanded View ($\gamma=2.5$)

From all these plots, it is clear that increasing compensator order does produce a better solution. However, after 9th order, no more significant improvement is seen. The question then becomes: is 3n an optimal order? Are the 12th and 18th order controllers simply nonminimal realizations of the 9th order controller? Several different approaches were taken to answer this question. The first approach was to examine the Hankel singular values of the compensators. If the 9-state controller truly has an optimal order, it might be expected that the 9-state controller would have 9 relatively signficant Hankel singular values. The 12-state controller would have 3 Hankel singular values that are either exactly zero or at least several orders of magnitude smaller than the other 9. Likewise, the 18-state contoller would have 9 relatively small Hankel singular values. However, this phenomenon was not found to be true. Table 5-3 shows a summary of the Hankel singular values of K_{mix} for each order, in rough magnitudes. Notice that even though some of the Hankel singular values get relatively small for the higher order compensators, there is never any definite division where the remainder of the states are clearly superfluous. Thus, at first glance, this approach does not give much help in determining if the higher order compensators should be reduced.

Next, the actual poles and zeros of the compensators were plotted in order to get a different look at possible pole/zero cancellations. Unfortunately, this did not provide any more help. Every compensator 6th order and up had multiple poles with zeros right on top of them. Simply by inspection, if the approximate

Table 5-3. Hankel Singular Values of K_{mix} (γ=2.5)

3 state	4 state	5 state	6 state	7 state	8 state	9 state	12 state	18 state
7e-0	7e-0	7e-0	6e-0	7e-0	3e-0	4e-0	6e-0	5e-0
6e-0	6e-0	6e-0	6e-0	6e-0	3e-0	4e-0	5e-0	5e-0
2e-0	2e-0	2e-0	2e-0	2e-0	3e-0	3e-0	3e-0	5e-0
	1e-1	2e-1	1e-1	1e-1	3e-0	3e-0	1e-0	3e-0
		3e-2	4e-2	7e-2	2e-0	2e-0	9e-1	2e-0
			3e-2	4e-2	4e-1	1e-1	2e-1	1e-1
				3e-3	5e-2	5e-2	7e-2	5e-2
					4e-2	4e-2	5e-2	4e-2
						2e-3	4e-2	1e-2
							2e-2	1e-2
							9e-3	6e-3
							3e-4	4e-4
								2e-4
								7e-5
								3e-5
								3e-5
								4e-6
								4e-7

pole/zero cancellations are made, each of the higher order compensators would reduce to

6-state $\rightarrow$ 4-state

7-state $\rightarrow$ 4-state

8-state $\rightarrow$ 4-state

9-state $\rightarrow$ (4 or 6)-state

12-state $\rightarrow$ (7 or 8)-state

18-state $\rightarrow$ (8 or 10)-state

The final approach that was tried was to simply perform a Schur balanced model order reduction on the 12 and 18-state compensators to see what they looked like when reduced to 9th order. It was quickly discovered, however, that it is impossible to remove even one state without immediately violating the ∞-norm bound on T_{ed}. This means that these reduced order controllers are no longer valid solutions to the mixed problem. It is not really surprising that this happens because the solution (for each order) lies right on the boundary of the ∞-norm constraint. Therefore, any reduction in order immediately makes the solution unacceptable. Based on this result, it seems that all the apparent pole/zero cancellations are not true cancellations. These additional poles may have very small residues, but they need to remain in the solution.

5.3.2 γ=3.0 Results. In order to see if the same trends observed in the γ=2.5 case are typical, the γ level was fixed at 3.0. This time, however, only 6, 9, and 12-state solutions were found. Table 5-4 shows a summary of the higher order results for γ = 3.0.

Table 5-4. Higher Order Results, γ=3.0

Compensator Order	α^*	γ^*
3 (full)	10.460	3.0
6	10.44380	3.0
9	10.42787	3.0
12	10.42619	3.0

Figure 5-24 shows this same data graphically.

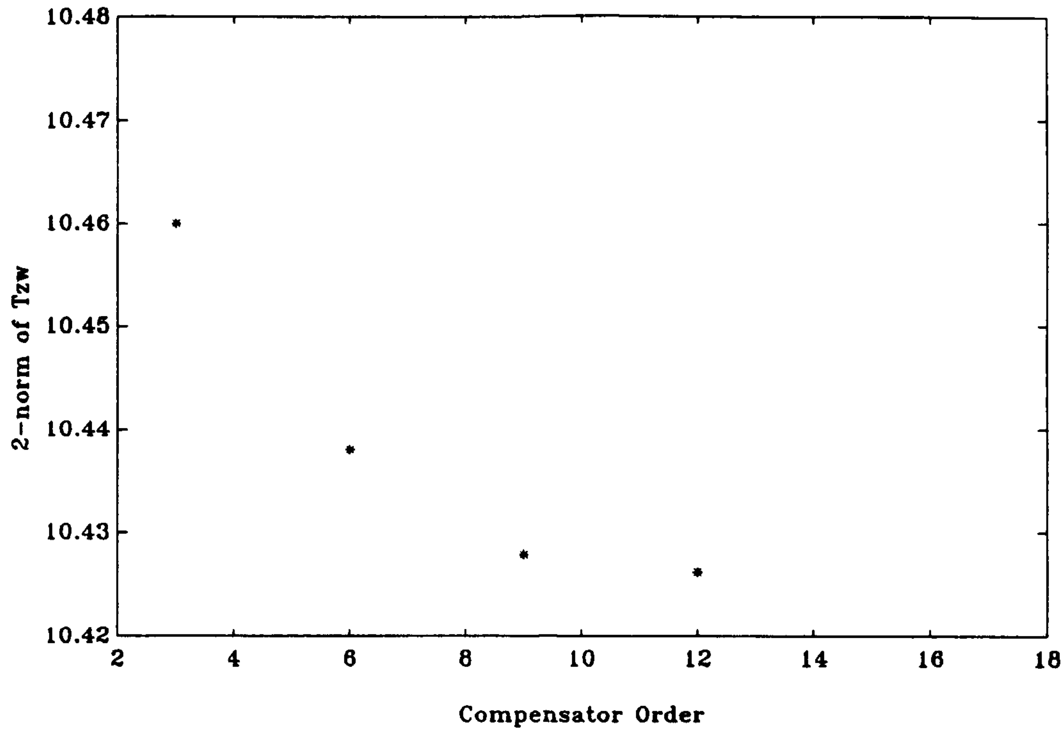

Figure 5-24. Higher Order Results, $\gamma=3.0$

The same decrease in α^* for increased order compensators is clearly seen for this level of γ also. Notice, however, that there is less of an improvement than for the $\gamma=2.5$ case. This is because the full order mixed solution is already closer to the H_2 optimal solution due to the relaxation of γ. Figures 5-25, 5-26, and 5-27 show the singular value plots of K_{mix}, T_{zw}, and T_{ed} for orders 6, 9 and 12. The same trends that were observed for $\gamma=2.5$ are apparent here. The recovery of the H_2 solution is again seen. As shown in Figure 5-28, the $\gamma=3.0$ solution is able to recover more of the H_2 solution due to the higher level of γ. Also, the higher order compensators do a better job with this recovery.

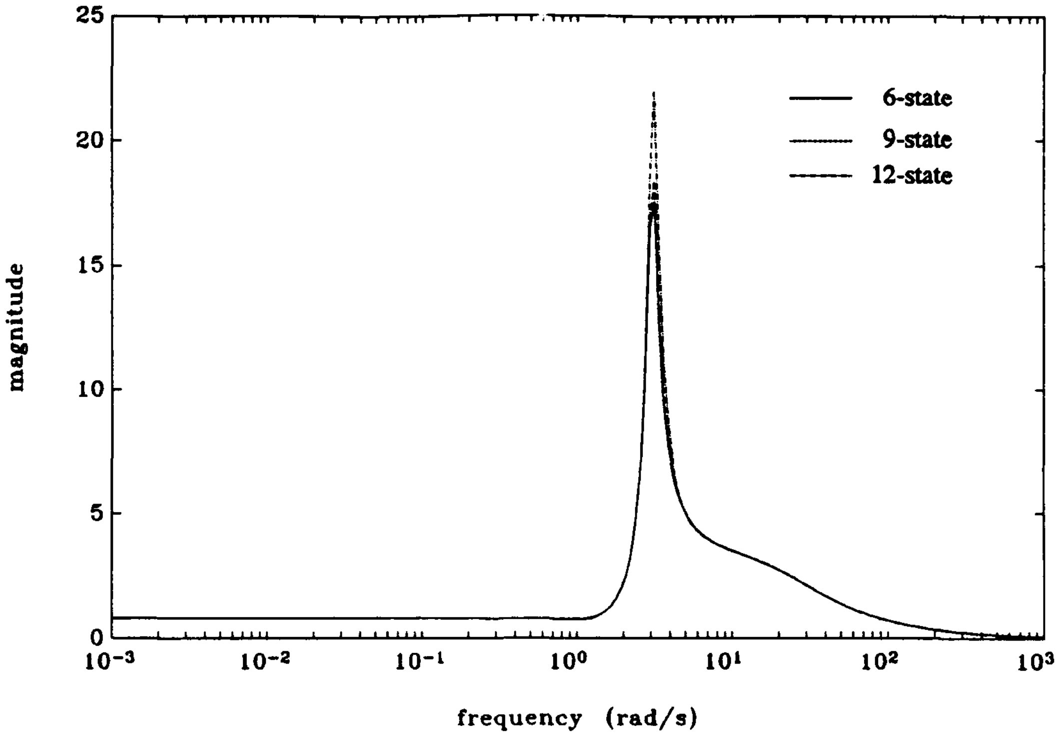

Figure 5-25. Singular Value Plot of K_{mix}, $\gamma=3.0$
(6,9,12-state)

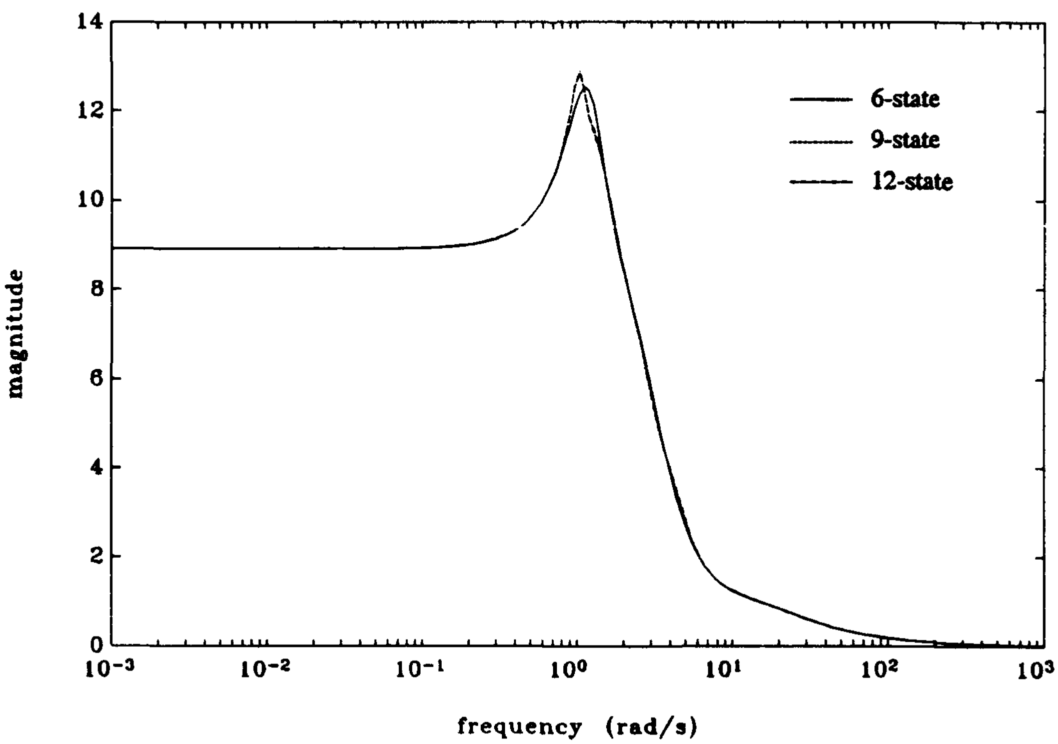

Figure 5-26. Singular Value Plot of T_{zw}, $\gamma=3.0$
(6,9,12-state)

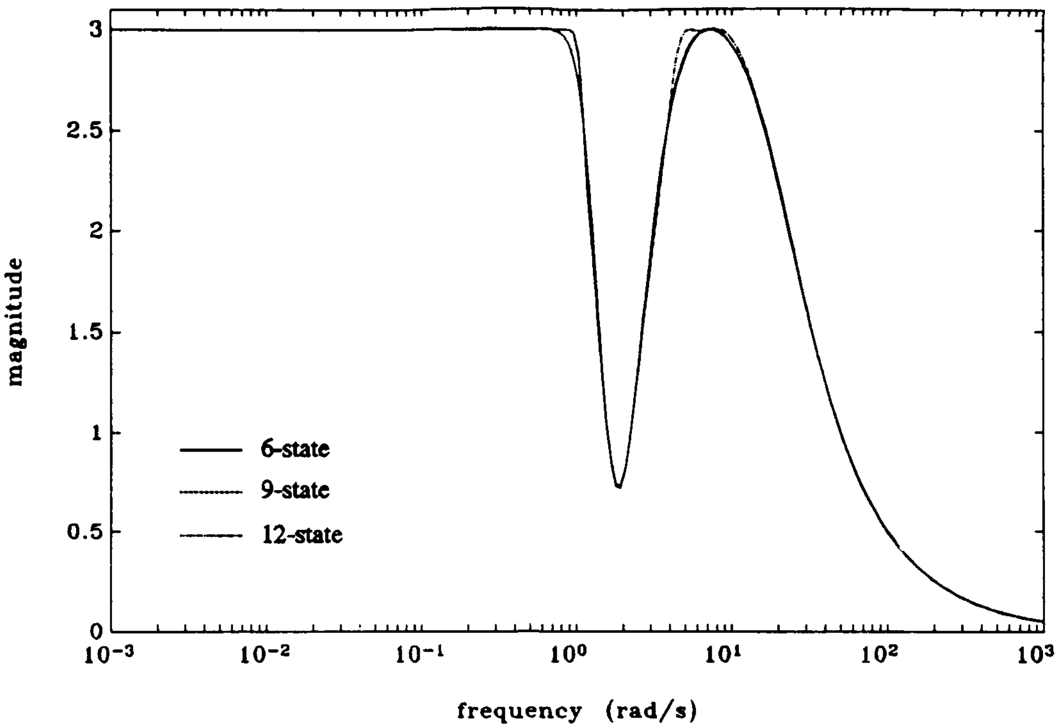

Figure 5-27. Singular Value Plot of T_{ed}, $\gamma=3.0$
(6,9,12-state)

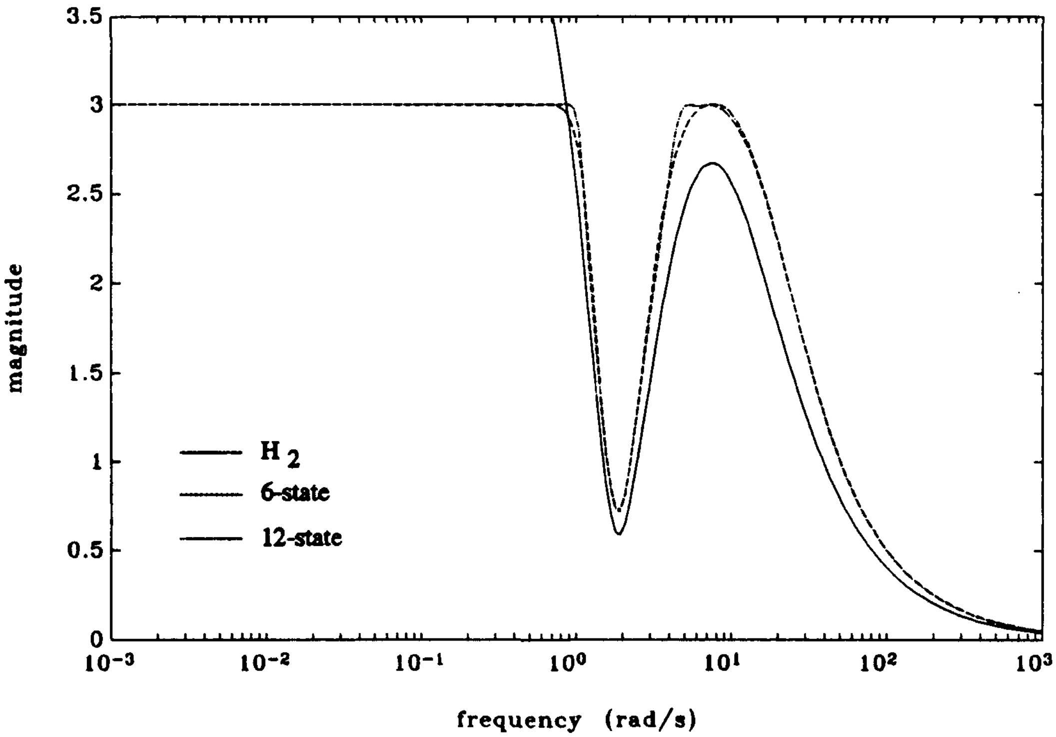

Figure 5-28. T_{ed} Comparison Plot for $\gamma=3.0$

5.3.3 Ninth-Order Results, γ Sweep. Now, consider freezing the compensator order and finding the solutions for a whole range of γ's. Since the 9-state controllers for the two γ levels examined so far appear to be near the limit of achievable mixed performance, select the compensator order to be 9. Table 5-5 shows a summary of the results.

Figure 5-5. Results of 9-state γ Sweep

γ	3-state K_{mix} α^*	9-state K_{mix} α^*	Percent Difference
2.2	13.868	13.697	1.230
2.25	13.295	13.085	1.578
2.35	12.501	12.243	2.060
2.5	11.616	11.488	1.105
2.75	10.870	10.795	0.690
3.0	10.460	10.428	0.307
3.25	10.225	10.219	0.059
3.5	10.086	10.078	0.079

Note that the 3-state solutions are taken directly from Table 5-1. Figure 5-29 shows Ridgley's full order mixed plot with the 9th order solutions superimposed. Figure 5-30 is an expanded view of this same plot. One fact is very obvious: the 9th order compensators do not give large improvements in terms of reducing the 2-norm for this example. The largest decrease in α^* is about 2%.

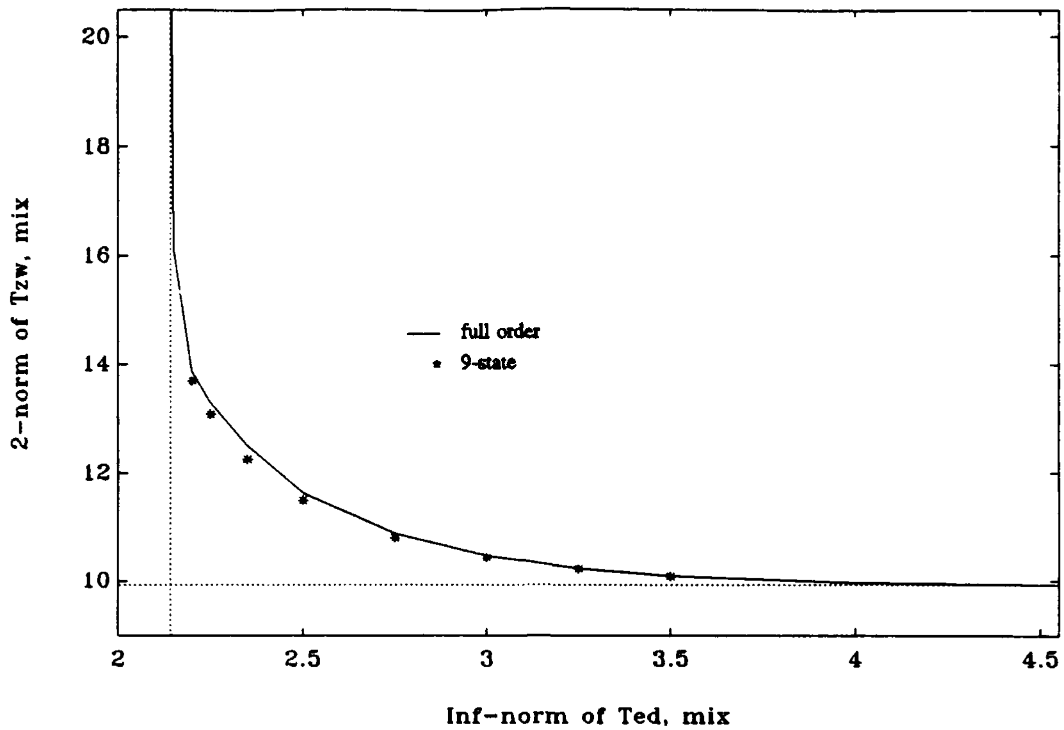

Figure 5-29. Mixed Plot, 3 and 9-state Comparison

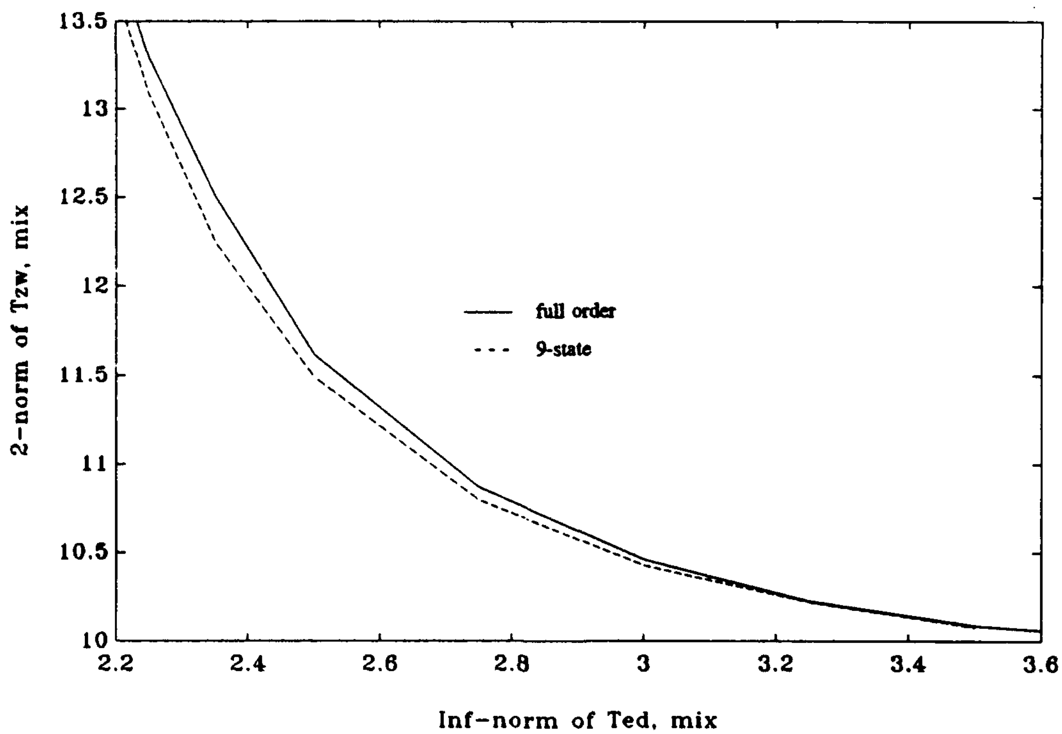

Figure 5-30. Mixed Plot (expanded), 3 and 9-state Comparison

The mixed solution singular value plots of T_{ed} and T_{zw} for the 9th order compensators are given in Figures 5-31 and 5-32. These plots show a definite recovery of the H_2 optimal solution. This was true for the full order case [Rid91a,164], and is seen to be true for the higher order case as well. In fact, as shown earlier, the higher order compensators actually do a better job of recovering the unconstrained H_2 optimal solution than the full order controllers.

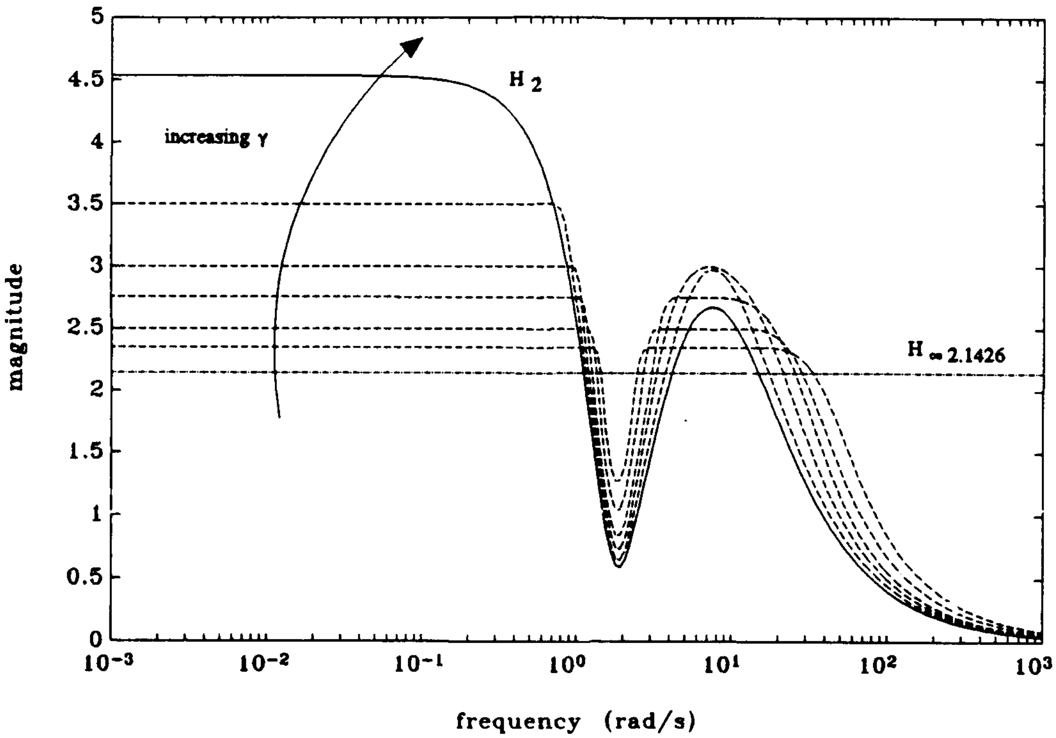

Figure 5-31. Mixed T_{ed} Plot (9-state)
γ = 2.1426, 2.25, 2.5, 2.75, 3.0, 3.5, 4.5364

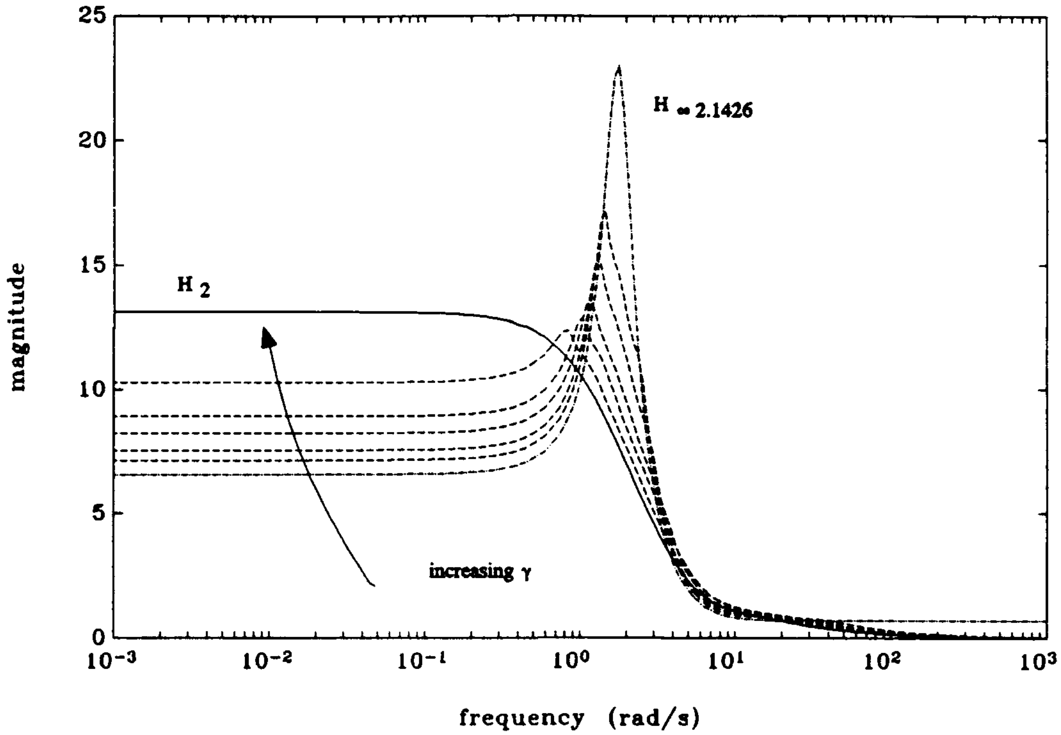

Figure 5-32. Mixed T_{zw} Plot (9-state)
γ = 2.1426, 2.25, 2.5, 2.75, 3.0, 3.5, 4.5364

5.4 MIMO Mixed Optimization Example

Consider the mixed H_2/H_∞ optimization system block diagram shown in Figure 5-3. For this multi-input multi-output example, all signals (d, w, e, z, u and y) are assumed to be two-dimensional vectors. In order to make direct comparisons with a known full order case, the same system that was defined in [Rid91a,170-171] was chosen for this analysis. The state space matrices of the system are:

$$A = \begin{bmatrix} -5 & 2 & 14 & 20 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.03 & 0.008 \\ 0.05 & 0.38 \\ 0.53 & 0.07 \\ 0.67 & 0.42 \end{bmatrix} \qquad B_w = \begin{bmatrix} 0.22 & 0.93 \\ 0.05 & 0.38 \\ 0.68 & 0.52 \\ 0.58 & 0.83 \end{bmatrix} \qquad B_u = \begin{bmatrix} 0.07 & 0.44 \\ 0.63 & 0.77 \\ 0.88 & 0.48 \\ 0.27 & 0.24 \end{bmatrix}$$

$$C_e = \begin{bmatrix} 0.55 & 0.33 & 1.80 & 0.12 \\ 0.72 & 0.97 & 1.82 & 1.81 \end{bmatrix}$$

$$C_z = \begin{bmatrix} 0.07 & 0.38 & 0.91 & 0.46 \\ 0.50 & 0.28 & 0.53 & 0.94 \end{bmatrix}$$

$$C_y = \begin{bmatrix} 0.05 & 0.77 & 0.13 & 0.69 \\ 0.76 & 0.83 & 0.02 & 0.87 \end{bmatrix}$$

$$D_{ed} = D_{ew} = D_{zd} = D_{zw} = D_{yu} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$D_{zu} = D_{yd} = D_{yw} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D_{eu} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

Note that this system does satisfy all the assumptions given in Section 3.1 This is a fourth order MIMO system whose "unweighted" plant, given by $P_{yu}(s) = C_y(sI-A)^{-1}B_u + D_{yu}$, is open-loop unstable and nonminimum phase. The singular value plot of P_{yu} is shown in Figure 5-33.

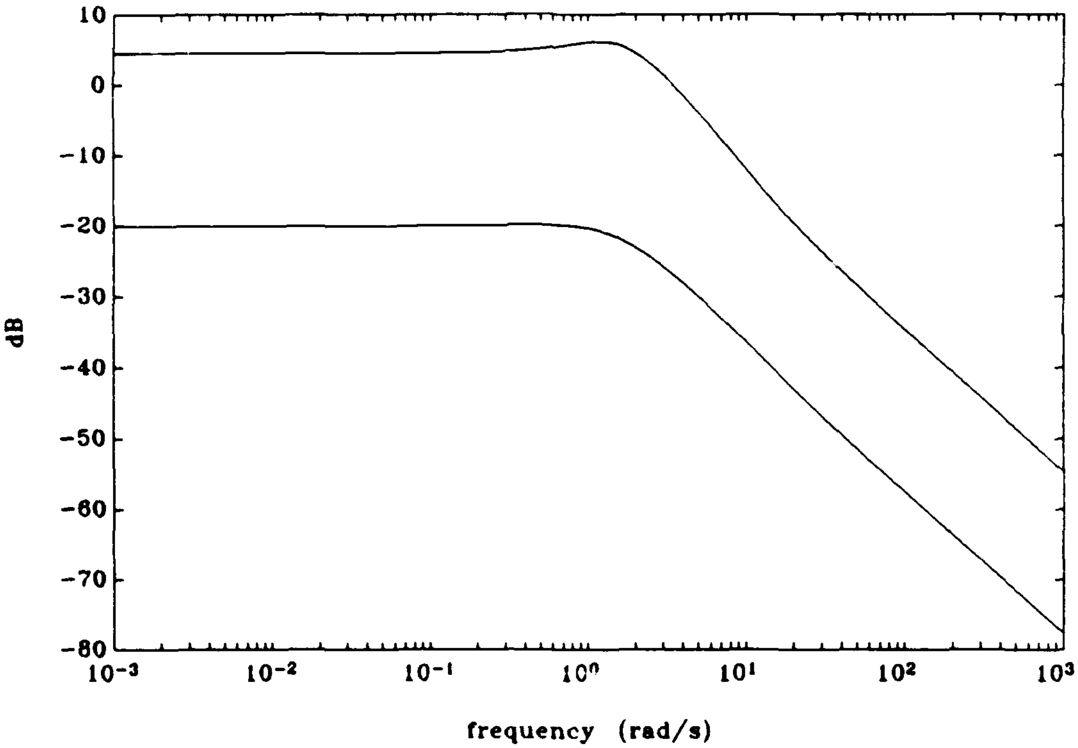

Figure 5-33. Singular Value Plot of MIMO Plant

Before beginning the mixed problem, it will be helpful to know the limits of achievable H_2 and H_∞ performance. Consider performing pure unconstrained H_2 and H_∞ optimization on the given plant. Figure 5-34 shows the singular value plot of the unique four-state H_2 optimal controller K_{2opt}. Figures 5-35 and 5-36 are the singular value plots of the corresponding closed-loop transfer functions T_{zw} and T_{ed}. The minimum achievable 2-norm of T_{zw} is $\alpha_o = 0.7975$. The ∞-norm of T_{ed} using K_{2opt} is $\gamma_2 = 40.548$.

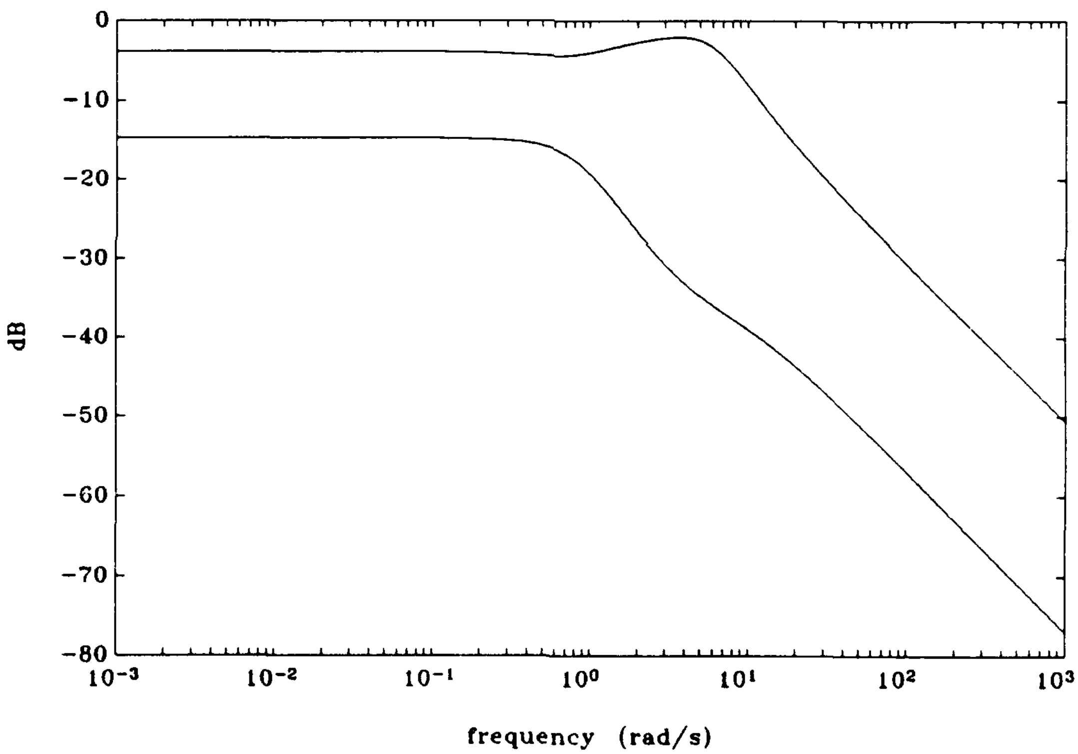

Figure 5-34. Singular Value Plot of K_{2opt}

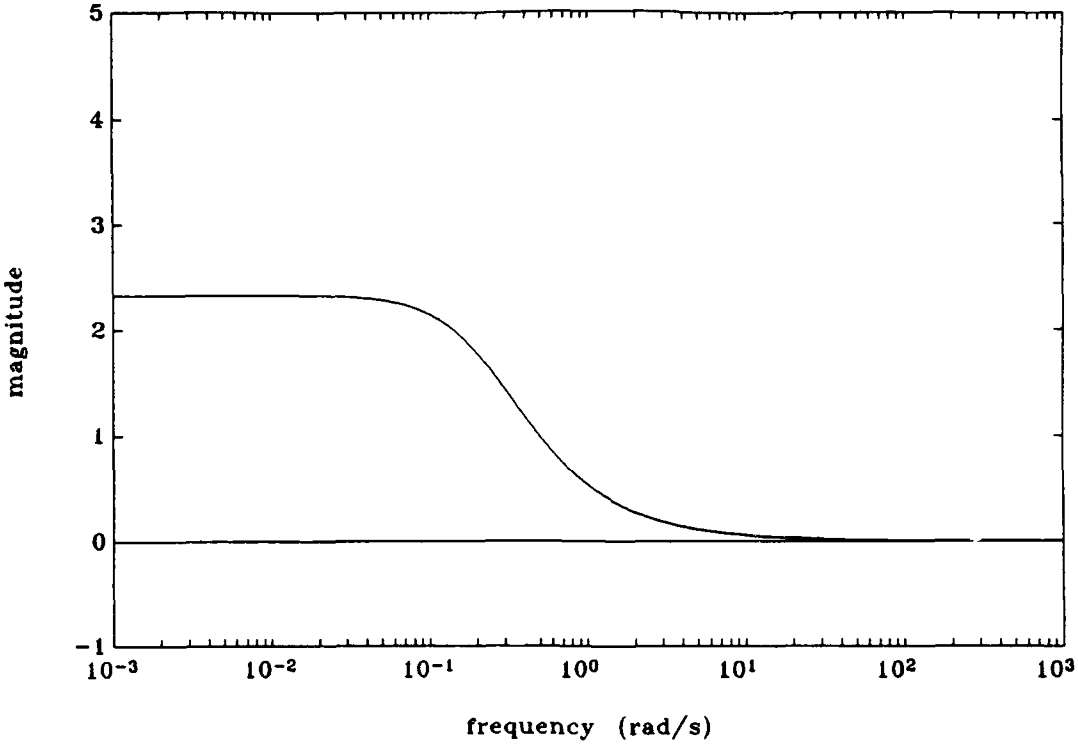

Figure 5-35. Singular Value Plot of T_{zw} for K_{2opt}

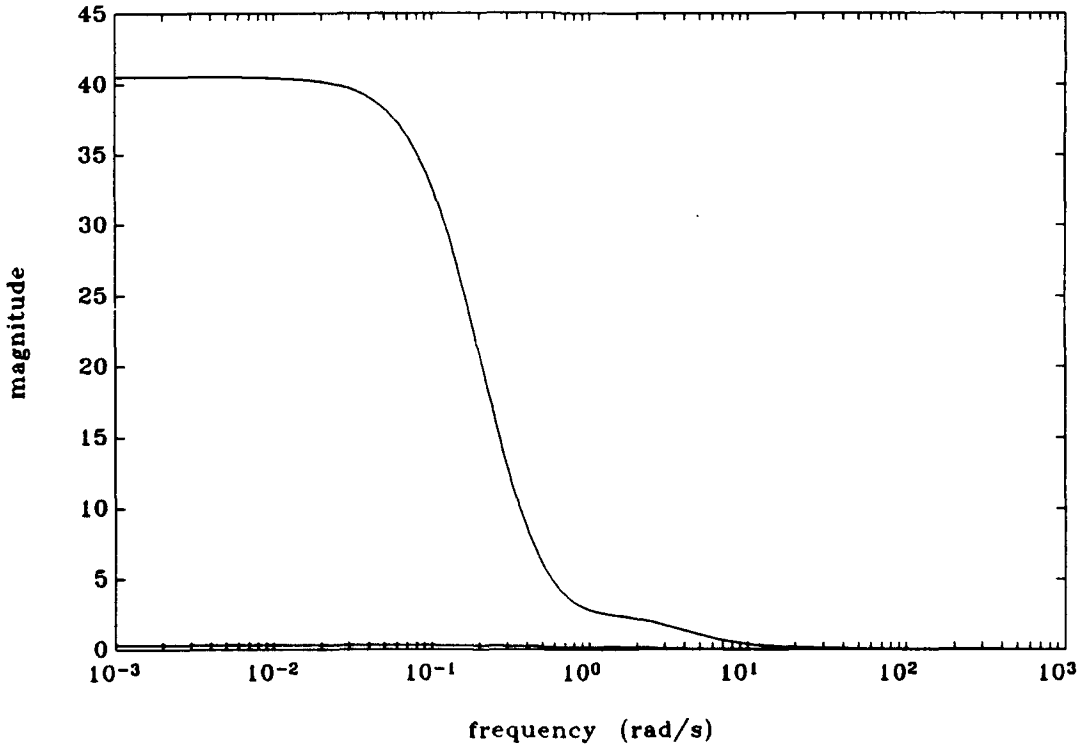

Figure 5-36. Singular Value Plot of T_{ed} for K_{2opt}

Now, if H_∞ optimization is performed, the minimum achievable ∞-norm of T_{ed} is found to be about $\gamma_o \approx 2.3012$. The freedom parameter Q(s) from the parameterization of H_∞ controllers must be specified. The central H_∞ compensator is of particular interest since it will be the initial guess for the DFP program (at $\mu = 1$), so choose $Q(s) = 0$. The singular value plot of the H_∞ suboptimal central compensator for $\gamma = 2.3012$ ($K_{\infty 2.3012}$) is given in Figure 5-37. Figures 5-38 and 5-39 are the singular value plots of the corresponding closed-loop transfer functions T_{zw} and T_{ed}. The 2-norm of T_{zw} for $K_{\infty 2.3012}$ is $\|T_{zw}\|_2 = 1887.3$, considerably higher than α_o.

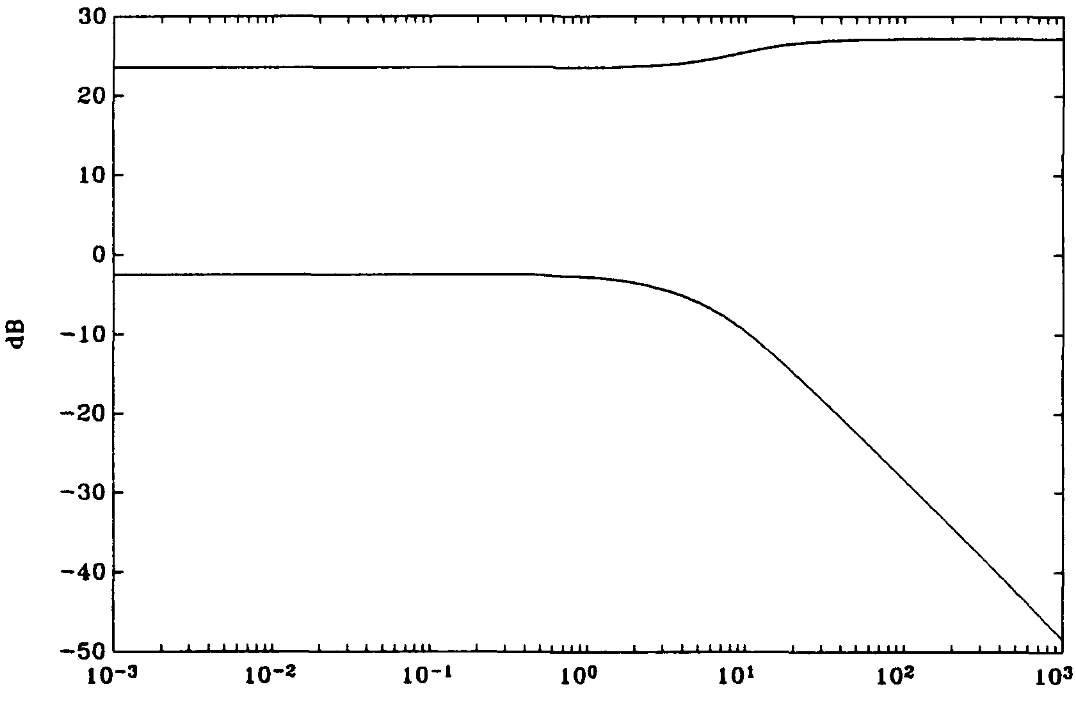

Figure 5-37. Singular Value Plot of $K_{\infty 2.3012}$

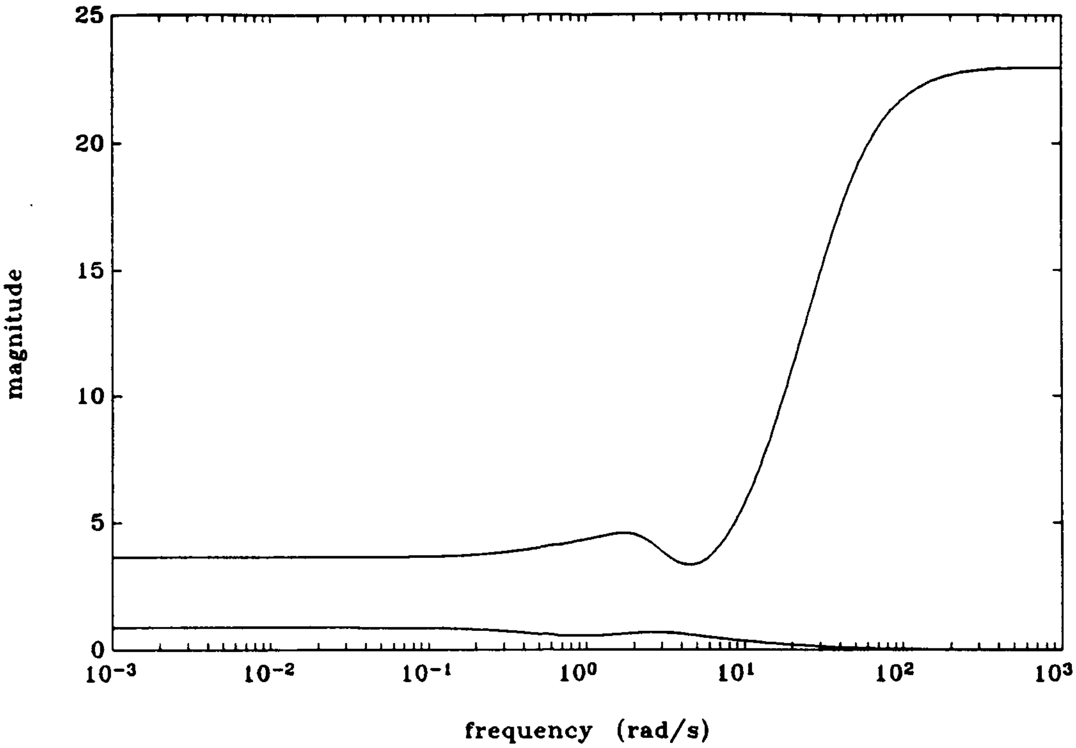

Figure 5-38. Singular Value Plot of T_{zw} for $K_{\infty 2.3012}$

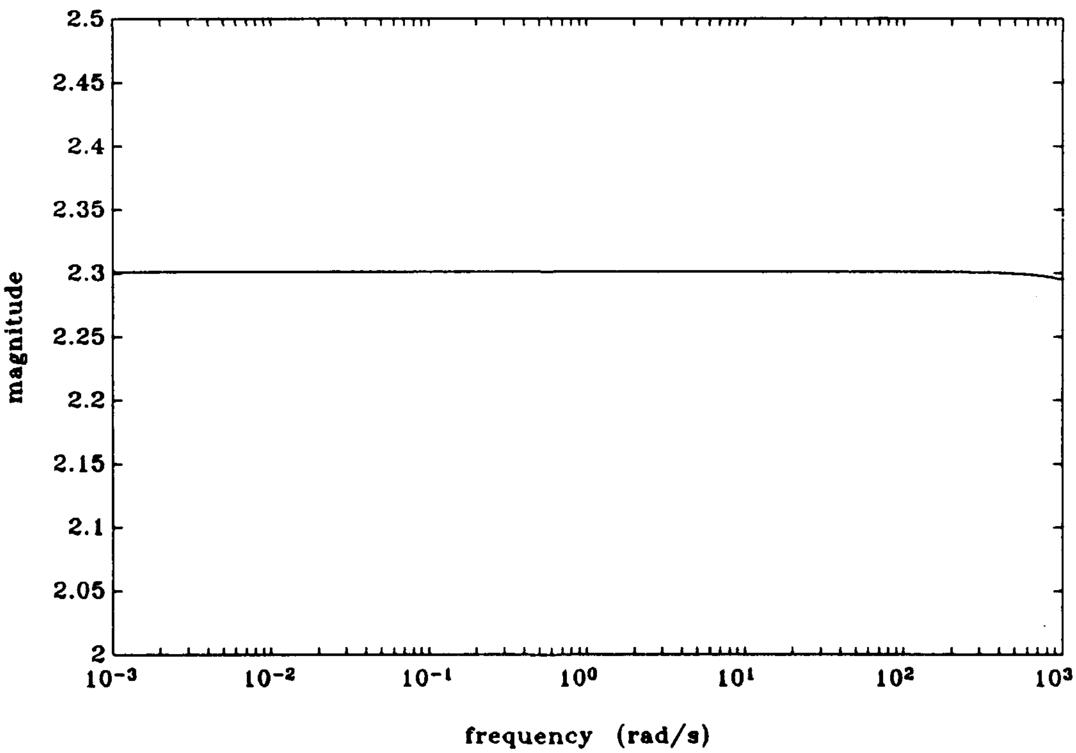

Figure 5-39. Singular Value Plot of T_{ed} for $K_{\infty 2.3012}$

Although it is not evident in these plots, they all have a high frequency roll off. If the optimal H_∞ controller was found, there would be no roll off and $\|T_{zw}\|_2$ would be infinite.

Now consider performing mixed H_2/H_∞ optimization, assuming the controller is full order. A brief summary of some of Ridgely's MIMO results are shown in Table 5-6 and Figure 5-40. Table 5-6 shows the values of $\|T_{zw}\|_2$ and $\|T_{ed}\|_\infty$ for the mixed and central H_∞ controllers at varying levels of γ. Notice that $\|T_{ed}\|_\infty = \gamma$ for the mixed controller. Actually, the ∞-norm values are rounded off (typically within 0.00001). Figure 5-40 shows this data in graphical form. Note that γ_2 is not shown on the plot because it is out at 40.548. This shows the tremendous value of the mixed controllers. The long, flat "tail" on the mixed curve enables huge reductions in the ∞-norm of T_{ed} while making only small sacrifices in the 2-norm of T_{zw}. No further discussion of these full order results will be given here. Rather, these results are included as a point of reference for the higher order results.

Table 5-6. MIMO Example Full Order Results

γ	(Q = 0) $\lVert T_{ed} \rVert_\infty$	(Q = 0) $\lVert T_{zw} \rVert_2$	(mix) $\lVert T_{ed} \rVert_\infty$	(mix) $\lVert T_{zw} \rVert_2$
2.3012	2.3012	1887.3	2.3012	~ 1887.3
2.32	2.3199	119.129	2.32	115.713
2.35	2.3495	62.307	2.35	47.619
2.4	2.3979	37.008	2.4	23.664
2.5	2.4914	22.155	2.5	10.713
2.6	2.5807	16.799	2.6	6.9406
2.75	2.7069	13.11	2.75	4.6072
3.0	2.8975	10.415	3.0	3.1622
4.0	3.4578	7.5491	4.0	1.9115
5.0	3.7978	6.8949	5.0	1.4582
10.0	4.3708	6.2236	10.0	0.9800
40.548	4.5874	6.0753	40.548	0.7975
50.0	4.5925	6.0723	40.548	0.7975
100.0	4.5997	6.0678	40.548	0.7975

[Rid91a,188]

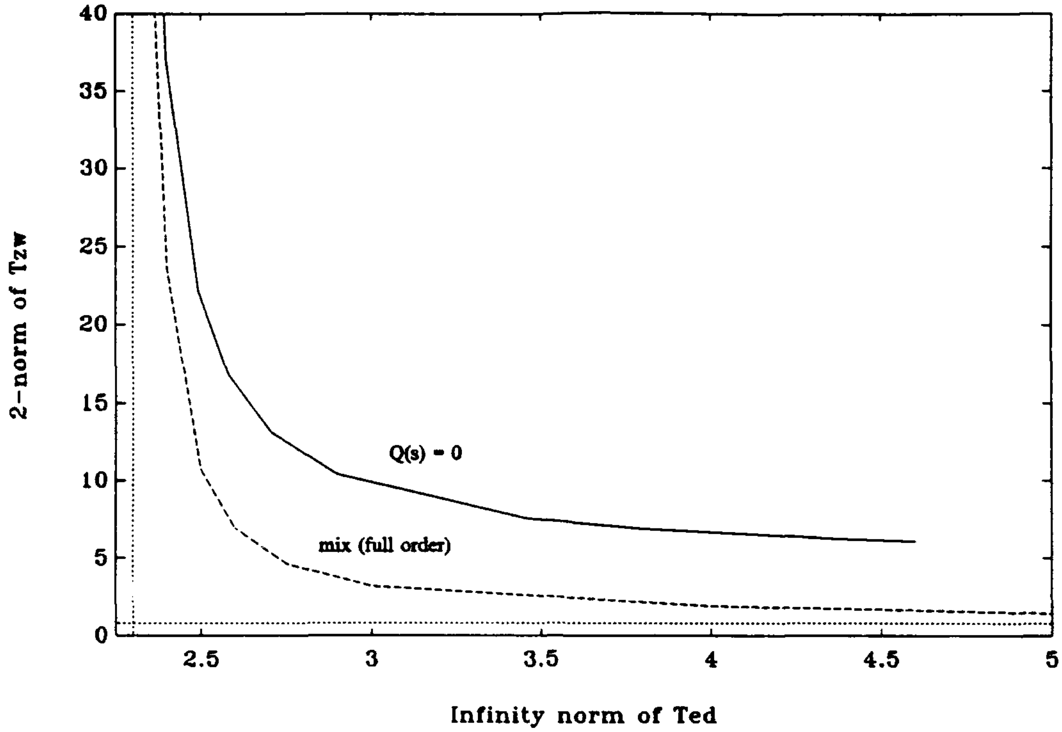

Figure 5-40. MIMO Example Full Order and Q(s)=0 Results

[Rid91a,189]

Now, consider the case of higher order compensators. As in the SISO example, the basic approach in the MIMO problem was to begin with the full order case and continually increase the compensator order until some kind of trend could be recognized. Before this order sweep could be accomplished, the design γ had to be chosen. As γ approaches γ_0, the numerics become more and more difficult. Therefore, a level of $\gamma=3.0$ was selected. This was the only γ that was run due to the excessive computer time required to find a solution (for example, the entries in Table 5-7 typically took from about eight to twelve hours for the 4-state solution to about two weeks of almost continuous run-time for the

16-state solution). In retrospect, perhaps a slightly lower γ might have been a better choice to demonstrate the value of the higher order compensators. As it turned out, the improvements due to the higher order controllers were observable, but they were small. Compensators of the following orders were then obtained by running DFP: 4, 6, 7, 8, 9, 10, 12, 16.

Table 5-7 shows a summary of the higher order results for $\gamma = 2.5$. Note that the ∞-norm values are essentially the same as γ. They are actually slightly less than γ (typically by about 10^{-6}).

Table 5-7. Higher Order Results, γ=3.0

Compensator Order	α^*	γ^*
4 (full)	3.15799	3.0
6	3.15530	3.0
7	3.15411	3.0
8	3.15246	3.0
9	3.15238	3.0
10	3.15232	3.0
12	3.15231	3.0
16	3.15216	3.0

Figure 5-12 shows this same data on a graph.

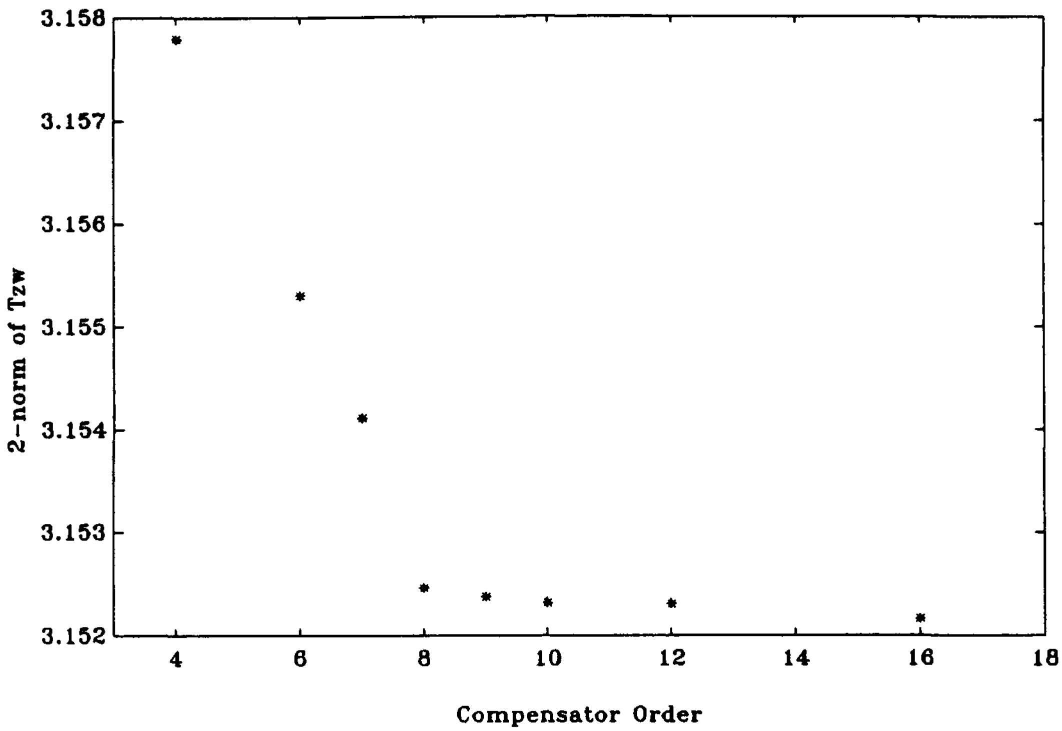

Figure 5-41. Higher Order Results, γ=3.0

One observation can immediately be made: in general, the optimal order is definitely not the order of the plant. This graph clearly shows that as order is increased beyond full order (n=4), the value of the 2-norm of T_{zw} decreases. This plot does not have a nice exponential-type decay as seen in the SISO example. It might be that some of the solutions are not completely converged. On the other hand, there is nothing that says what the shape of this curve must be. In fact, even though this curve is also strictly monotonically decreasing, there are no known proofs that guarantee this in general. The key observation here is that all the higher order compensators do have a lower α^* than the full order solution. Also, notice that this curve basically bottoms out at n_c = 8.

This is two times the order of the plant (recall that in the SISO example, this happened at three times the order of the plant).

The singular value plots of the mixed solutions are given in Figures 5-42 and 5-43. The corresponding singular value plots of T_{zw} are shown in Figures 5-44 and 5-45. Figures 5-46 and 5-47 are the corresponding singular value plots of T_{ed}. The same trends that were observed in the SISO case can be seen here in the MIMO case. As already mentioned, the higher order compensators do not produce large changes or improvements in this example. This is probably a combination of the system itself and the choice of γ. However, even though the changes are small, they do produce some improvement. In order to show the recovery of the mixed solution to the H_2 solution, Figure 5-48 shows an expanded view of the T_{ed} plots. The H_2 solution, 4-state mixed, and 16-state mixed solutions are given for comparison. Notice that even though the mixed solutions do not make the sharp turn to follow the H_2 curve immediately, they do make the sharp turn. The 16-state mixed compensator makes this turn sharper than the 4-state compensator (as seen in the SISO example).

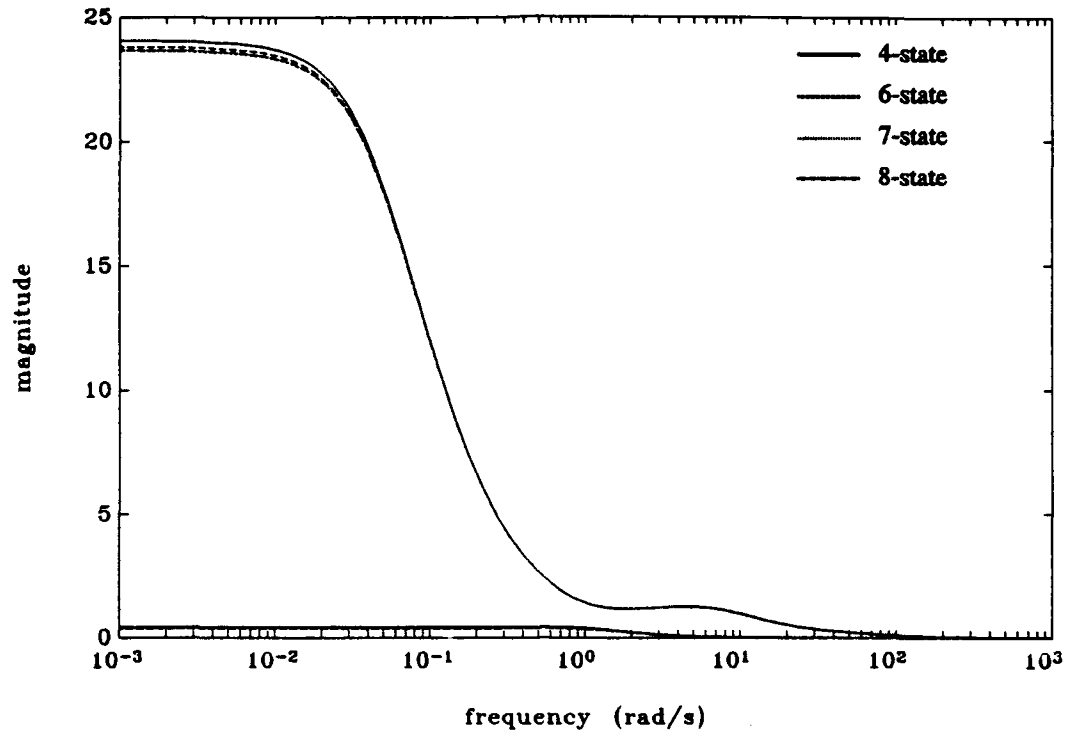

Figure 5-42. Singular Value Plots of K_{mix} (4,6,7,8-state)

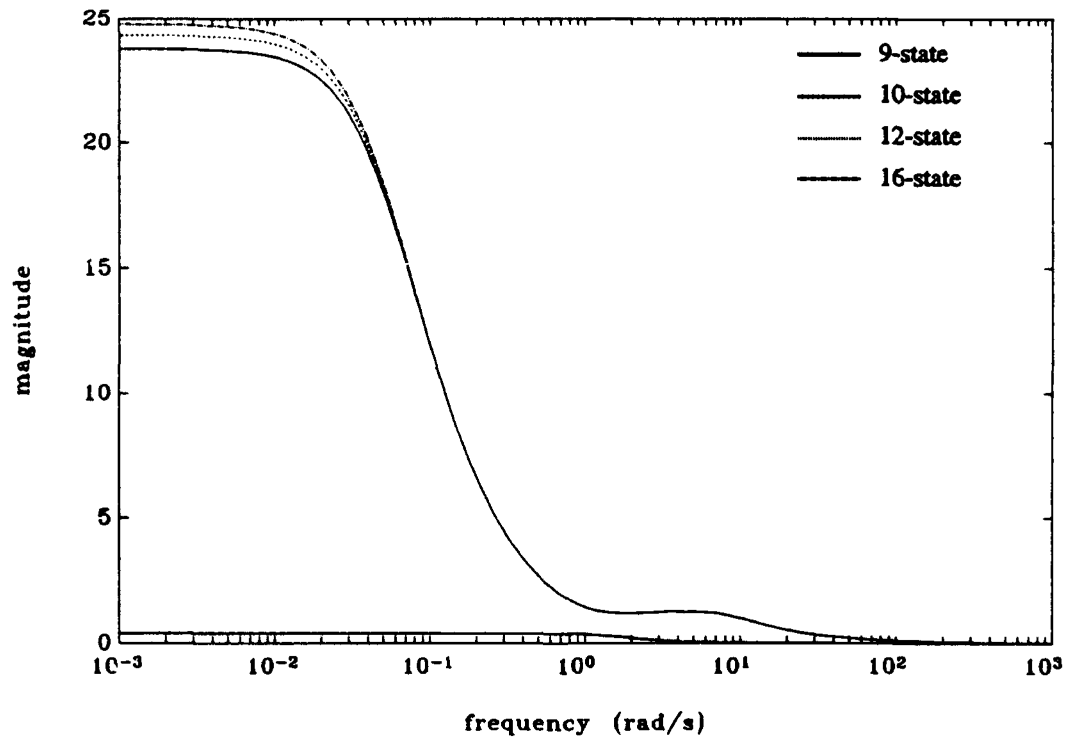

Figure 5-43. Singular Value Plots of K_{mix} (9,10,12,16-state)

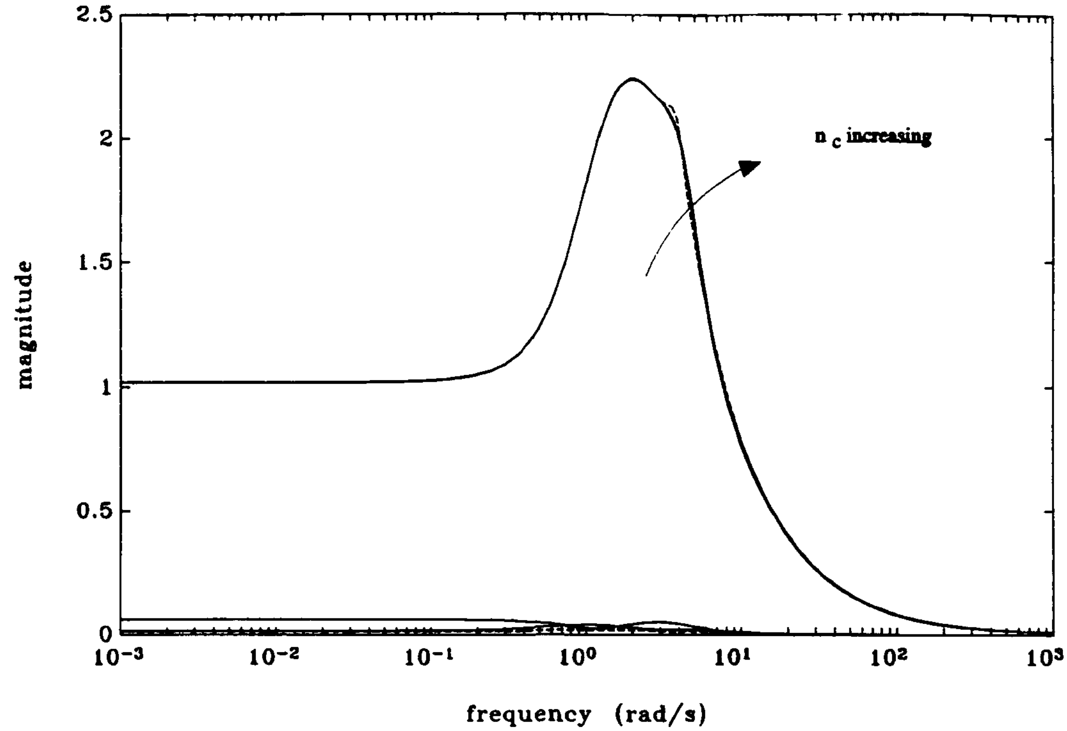

Figure 5-44. Singular Value Plots of T_{zw} for γ=3.0
(4,6,7,8-state)

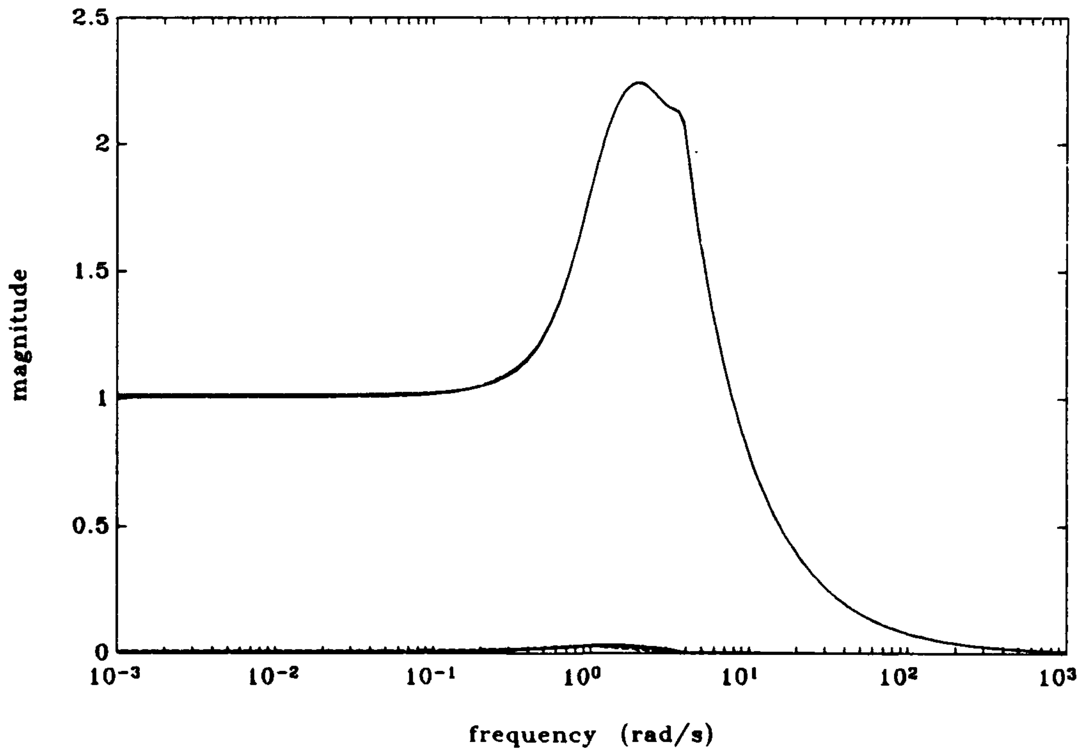

Figure 5-45. Singular Value Plots of T_{zw} for γ=3.0
(9,10,12,16-state)

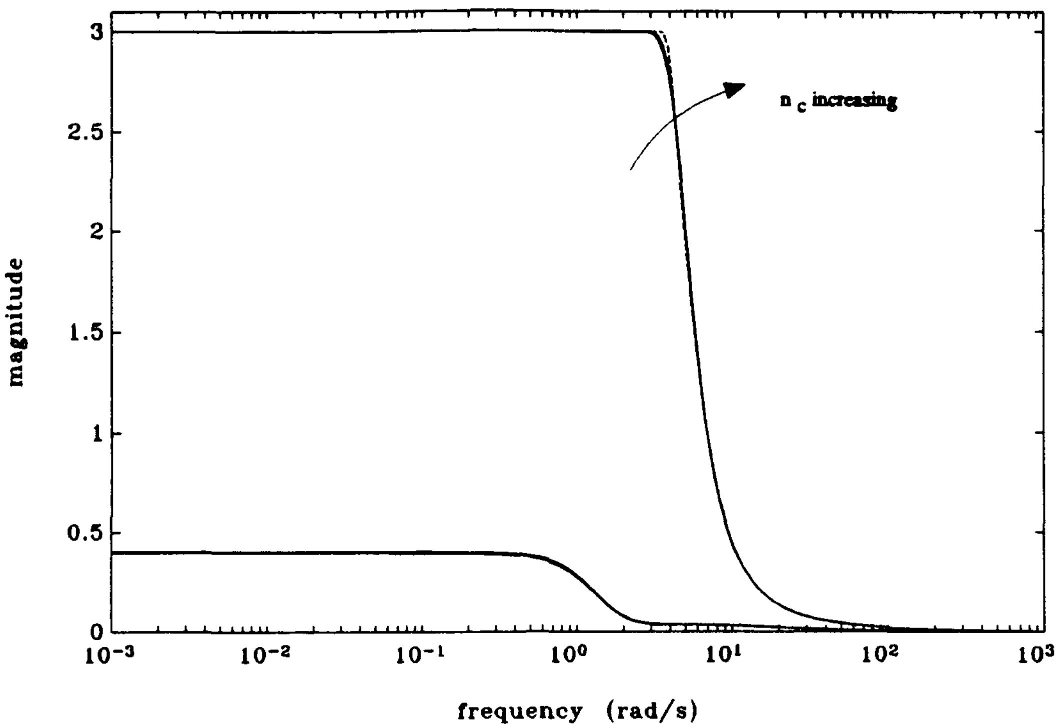

Figure 5-46. Singular Value Plots of T_{ed} for $\gamma=3.0$
(4,6,7,8-state)

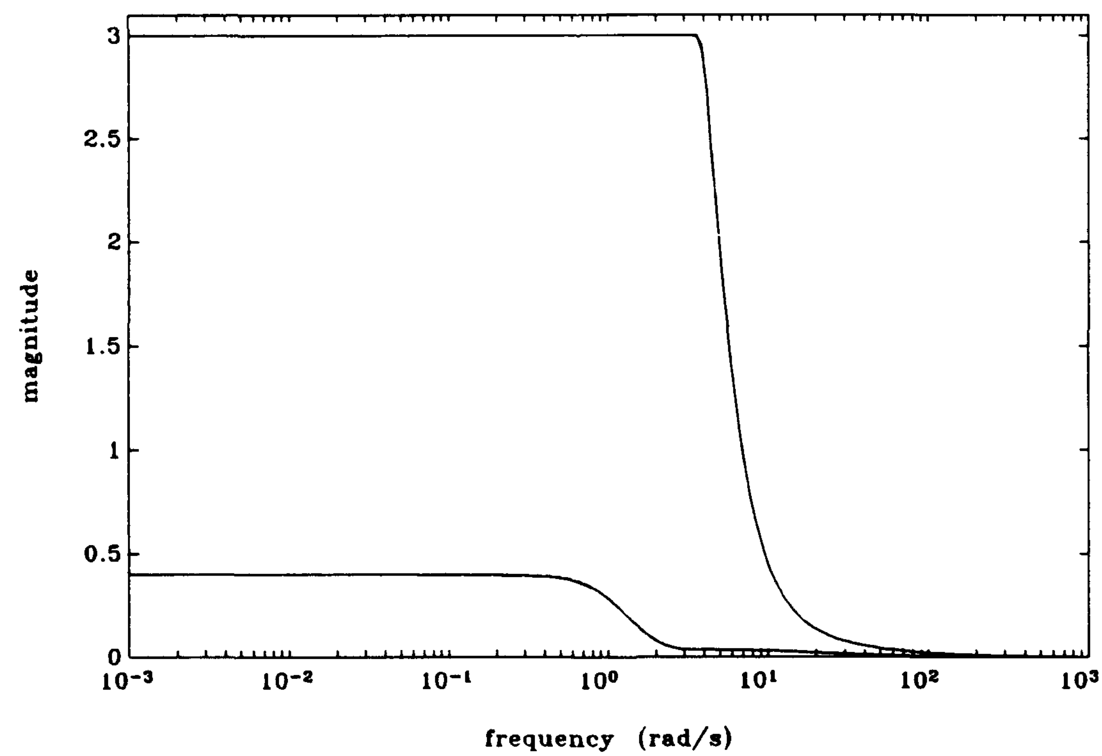

Figure 5-47. Singular Value Plots of T_{ed} for $\gamma=3.0$
(9,10,12,16-state)

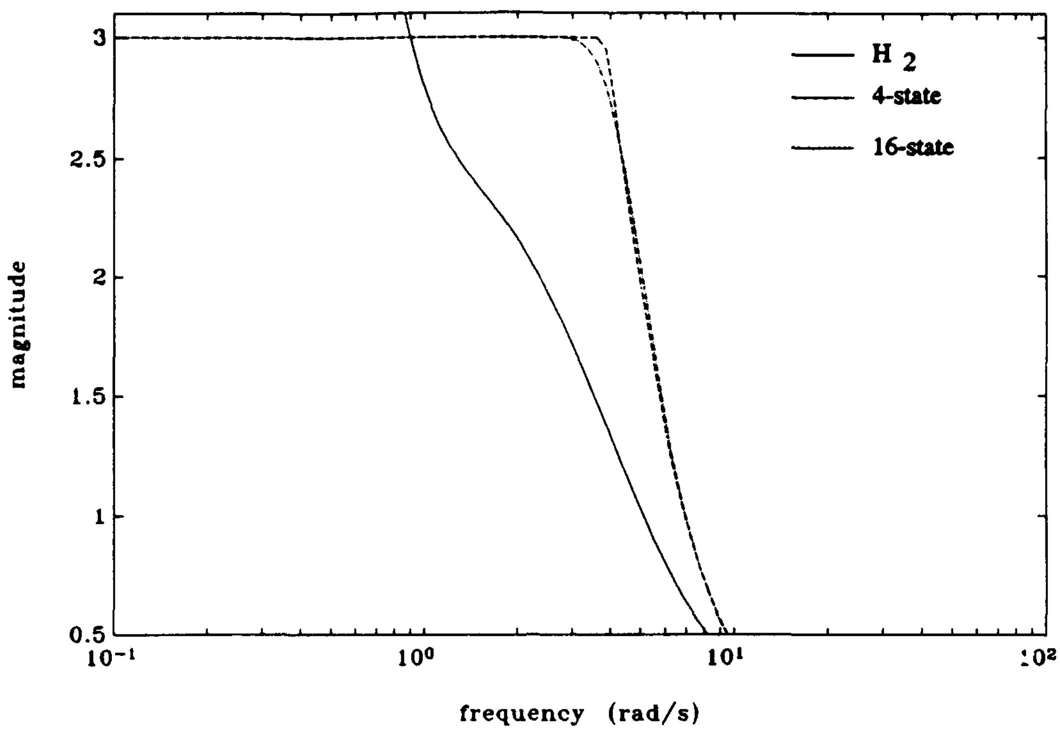

Figure 5-48. T_{ed} Comparison Plot Expanded View (γ=3.0)

VI. Conclusions and Recommendations

6.1 Optimal Order

Extensions from the full order case to the increased order case were made in Chapter IV, and it has been shown by numerous examples that the optimal order of the mixed H_2/H_∞ solution is not, in general, the order of the plant. This is a departure from the nature of the separate, unconstrained H_2 and H_∞ problems, and only serves to demonstrate the complexity of the mixed problem. While it is now known that the optimal order of the mixed problem is not, in general, the order of the plant, the ultimate question of what the optimal order is remains to be proven.

At the outset, it was believed that the optimal order might be 3n (and some of the initial findings seemed to confirm this.) A brief outline of the rationale for this conjecture is as follows. Since the ∞-norm bound must be satisfied, the compensator K(s) can be parameterized as a lower LFT of J and Q as shown in Figure 6-1, where J is given by an H_∞ parameterization [DGKF89] and $Q \in RH_\infty$, $\|Q\|_\infty \leq \gamma$.

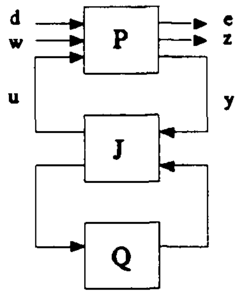

Figure 6-1. Mixed Optimization Block Diagram with Q(s)

Since J is completely known, it can be combined with the nominal plant P to form a new plant P_J whose order is 2n (assuming no pole-zero cancellations occur). This is shown in Figure 6-2.

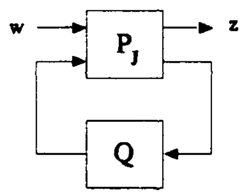

Figure 6-2. Closure of the P-J Loop Through H_∞ Optimization

The problem is now

$$\inf_{Q \in RH_\infty} \|T_{zw}\|_2 \quad \text{subject to } \|Q\|_\infty \leq \gamma$$

If it can be shown that the optimal order of Q is the order of this new plant (2n), the resulting compensator would have order 3n. Unfortunately, the problem set up in this manner is no more tractable than the original. Therefore, no analytical solution to this problem has been found either (if it had, the general mixed problem would be solved). Also, this Q could be further parameterized by a J_2 and Q_2 from an H_2 parameterization on P_J, and a new "plant" of order 3n created. If it could be shown that this continues indefinitely, it would lead to an optimal mixed compensator of infinite order.

Since it has been shown that adding states to the controller does cause improvement, it seems intuitive that this may continue indefinitely while asymptotically approaching some minimum achievable α^*. In fact, as a conclusion of this research, this is formally conjectured and supported with three evidences.

Conjecture: Assume γ is selected such that $\gamma_o < \gamma < \gamma_2$. The optimal order of K_{mix} is infinite.

Evidence 1: In both the SISO and MIMO examples, every increase in order of K_{mix} produced a reduction in the 2-norm of T_{zw}. Now, it could be rightfully argued that these reductions may be within the noise level of the computational abilities of the computer. Therefore, the reductions in the 2-norm are not real. However, care was taken to perform all calculations in double precision. Also, it does seem significant that reductions occurred in every instance. The numbers being dealt with may be small, but they should not be completely discounted.

Evidence 2: It has been demonstrated that the mixed solution tries to recover to the H_2 solution while still meeting the ∞-norm constraint. This is seen most clearly on the T_{ed} singular value plots. Figure 5-23 is repeated below in Figure 6-3 for the sake of discussion.

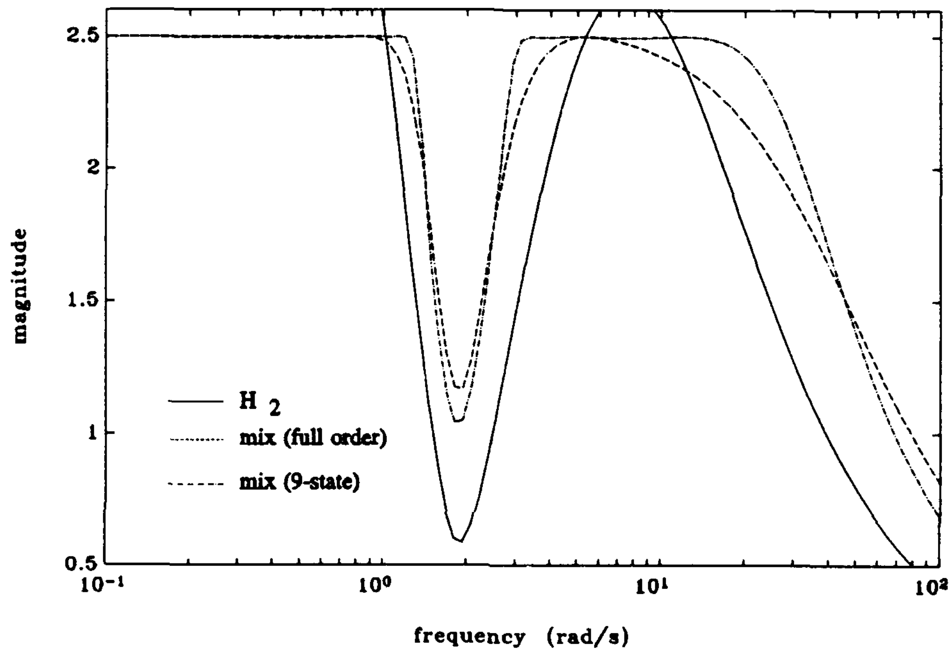

Figure 6-3. SISO T_{ed} Comparison Plot Expanded View (γ=2.5)

The mixed solution lies on the ∞-norm boundary wherever the H_2 solution is above the design γ level. When the H_2 solution drops below the ∞-norm bound, the mixed solution tries to follow this curve. Upon inspection of the higher order results, it appears that the point where the mixed curve turns to follow the H_2 curve, the true optimal mixed solution may have a point of discontinuity. Certainly, in all the examples, this turning point becomes sharper with increased compensator order. If it is true that this point is a point of discontinuity, it immediately follows that the compensator must have infinite order. The controller (at least for a SISO system) is nothing more than a ratio of polynomials in the Laplace domain. Therefore, in order to make this discontinuous turn, an infinite number of polynomial terms are required.

Evidence 3: Recall that two of the necessary conditions, Equations (3.10) and (3.11), are Lyapunov equations.

$$\tilde{A}\tilde{Q}_2 + \tilde{Q}_2\tilde{A}^T + \tilde{B}_w\tilde{B}_w^T = 0 \tag{6.1}$$

$$\tilde{A}^T X + X\tilde{A} + \tilde{C}_z^T\tilde{C}_z = 0 \tag{6.2}$$

Consider the following lemma:

Lemma 6.1: Suppose $X \geq 0$, $Z \geq 0$, $(\sqrt{Z}, A)$ is detectable and

$$A^T X + XA + Z = 0$$

Then A is stable.

Moreover, $X > 0$ iff $(\sqrt{Z}, A)$ is observable.

Proof: For the first conclusion, see ([Won85,283-284], Theorem 12.2). For the second conclusion, see ([Won85,58], Theorem 3.1).

Consider the SISO example. It was discovered that in every case examined, both X and $\tilde{Q}_2$ were positive definite. Therefore, by Lemma 6.1 and its dual, it follows that $(\tilde{A},\tilde{B}_u)$ is controllable and $(\tilde{C}_y,\tilde{A})$ is observable. In the SISO example, (A,B_u) is controllable and (C_y,A) is observable. It follows, then, that the compensator is also controllable and observable (i.e. it has a minimal realization). It was interesting to note, however, that while all the eigenvalues of X and $\tilde{Q}_2$ were positive and nonzero, usually only three had significant magnitude. Most of the rest of the eigenvalues were very close to zero. This corroborates with the earlier observation that many of the higher order compensator poles had zeros almost right on top of them and yet, due to ∞-norm bound, the pole/zero cancellations could not actually be made. If it is true that the mixed solution is indeed a minimal realization for orders up to infinity, it would follow that the optimal order of the mixed H_2/H_∞ solution is infinity.

6.2 Recommendations for Future Research

Obviously, the complete proof of optimal order remains to be shown. This work has shed some light on this subject, but it will require further research before a final answer can be given to the question of optimal order.

This research dealt completely with compensators whose order is greater than the order of the plant. As mentioned at the beginning of this thesis, the more practical research would be in the area of reduced order compensators. This is a much more difficult area of study, but it will be interesting to see the plot of

α^* versus compensator order completed by extending it below the order of the plant.

One issue that was looked at (but not resolved) is the problem of uniqueness of the solution. Recall that one of the necessary conditions in the general mixed problem is a Lyapunov equation with no constant term. The only way to get a nonzero solution to this equation is by requiring its "A" matrix to be neutrally stable. However, this leads to an infinite number of solutions of varying rank. It is not clear which of these solutions should be chosen or why the numerical solution converges to one solution over another. In terms of looking at the characteristics of higher order solutions, nonuniqueness of the solution does not cause much trouble. All the solutions that were obtained are valid and the plots shown in Chapter V are correct. However, when talking about optimal order, nonuniqueness of the solution is a problem. If there is a family of solutions to the mixed problem, it may be true that a different (full order or smaller increased order) compensator could achieve the same results as the higher order controllers shown here.

Research needs to continue into finding a closed-form solution to the mixed problem. It may be true that no closed-form solution exists. If this is the case, a more time efficient method for numerically solving the problem needs to be developed. Some of the higher order solutions literally took one to two weeks of almost continuous run time for a single solution. Before mixed H_2/H_∞ can become a useful design method from a practical standpoint, further research into solution algorithms is required.

6.3 Summary

In this thesis, the effects of using higher order compensators in mixed H_2/H_∞ have been investigated. The mixed problem in general has been discussed and the motivations for performing mixed H_2/H_∞ optimization have been presented. The issue of why higher order controllers are important was addressed and the optimal orders of related optimization problems were shown. Then, the general mixed problem was developed using a Lagrange multiplier technique. The necessary conditions for a minimum were derived and examined. Next, due to numerical difficulties in solving the true mixed problem, a suboptimal problem was developed and new necessary conditions given. After showing some key full order results, theoretical results for the higher order case were presented. In particular, the key proofs that were given include: the global minimum 2-norm is unachievable under output feedback for certain levels of γ regardless of compensator order; the solution to the mixed problem lies on the boundary of the ∞-norm constraint for this same range of γ's; and, the suboptimal mixed problem converges to the optimal in the limit for higher order compensators. Then, numerical SISO and MIMO examples were examined. It was seen that higher order compensators do produce a lower 2-norm and they are better able to recover to the H_2 solution than the full order controllers. Finally, based on the results from the numerical examples, it was conjectured that the optimal order of the mixed solution is infinite.

Appendix: FORTRAN Source Code of DFP Algorithm

This appendix contains the FORTRAN source code for the Davidon-Fletcher-Powell numerical algorithm used to solve the mixed H_2/H_∞ optimization problem. All code required to run the program is included here except for the Riccati equation solver and eigenvalue solver routines. These routines are readily available as public domain software. The calls to these routines, which will need to be modified depending on the software used, are in the subroutine EVALUF and are marked by ***bold italics***.

Below is a summary of the routines included:

```
C*********************************************************************
C
C  THIS FILE SHOULD BE 'INCLUDE'D IN THE DFP PROGRAM.  IT CONTAINS
C  ALL THE ARRAY SIZES NEEDED IN THE DFP DECLARATIONS.
C
C     ISTATE = LENGTH OF SYSTEM STATE VECTOR
C     KSTATE = LENGTH OF COMPENSATOR STATE VECTOR
C
C     NUMBD  = LENGTH OF EXOGENOUS INPUT VECTOR D
C     NUMBW  = LENGTH OF EXOGENOUS INPUT VECTOR W
C     NUMBU  = LENGTH OF CONTROLLED INPUT VECTOR U
C
C     NUMBE  = LENGTH OF CONTROLLED OUTPUT VECTOR E
C     NUMBZ  = LENGTH OF CONTROLLED OUTPUT VECTOR Z
C     NUMBY  = LENGTH OF MEASURED OUTPUT VECTOR Y
C*********************************************************************
C
       INTEGER   ISTATE,KSTATE,NUMBD,NUMBW,NUMBU,NUMBE,NUMBZ,NUMBY,
     .           TILDIM,NUMB
C
       PARAMETER (ISTATE =  3,
     .            KSTATE = 18,
     .            NUMBD  =  1,
     .            NUMBW  =  1,
     .            NUMBU  =  1,
     .            NUMBE  =  1,
     .            NUMBZ  =  1,
     .            NUMBY  =  1)
       PARAMETER
     .  (TILDIM =  ISTATE+KSTATE,
     .   NUMB   =  (KSTATE*KSTATE) + (NUMBU*KSTATE) +
     .                 (NUMBY*KSTATE))
C*********************************************************************
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                              c
c  This program uses the Davidon-Fletcher-Powell               c
c  numerical optimization algorithm to solve the               c
c  suboptimal mixed H2/Hinf/Entropy optimization               c
c  problem.  That is, determine (AC,BC,CC) that                c
c  minimizes the cost functional:                              c
c                                                              c
c  J = (1-AMU)*tr(Q2*CZ'*CZ) + AMU*tr(QINF*CE'*CE)             c
c                                                              c
c  where 0.0< AMU <1.0                                         c
c                                                              c
c  and Q2 is the real, symmetric, positive semidefinite        c
c  solution to the Lyapunov equation:                          c
c       ATIL*Q2 + Q2*ATIL' + BWTIL*BWTIL' = 0                  c
c                                                              c
c  and such that the Riccati equation:                         c
c    ATIL*QINF + QINF*ATIL' +                                  c
c      GAM2INV*QINF*CETIL'*CETIL*QINF + BDTIL*BDTIL' = 0       c
c  has a real, symmetric, positive semidefinite solution.      c
c                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      IMPLICIT REAL*8(A-H,O-Z)
c
c - INCLUDE the file containing all the array dimensions.
c   NOTE:  The file DIMEN.INC must be modified every time
c   there is a change in the size of the system or compensator,
c   and this program must be re-compiled.
c
      INCLUDE 'DIMEN.INC'
c
c - DFP algorithm variables
c
      COMMON/HMATRX/ H(NUMB,NUMB), S(NUMB), DELF(NUMB), DELOLF(NUMB)
      COMMON/FLAGS/  IFLAG2, ICNT, TOLCHK, CHECKSTOP
      COMMON/PARAM/ AMU, OMAMU, GAMMA, GAM2INV
c
c - Say hello and set the counters
c
      PRINT *,'THE GREAT AND MIGHTY DFP'
      PRINT *
      PRINT *,'Ahead one-half impulse power, Mr Crusher.'
      ICOUNT = 0
      JCOUNT = 0
      ISTOP  = 0
c
c - Open the input and output files
c
      OPEN(1,FILE='INPUT.DAT')
      OPEN(2,FILE='CHECK.DAT')
      OPEN(8,FILE='RESET.DAT')
      OPEN(9,FILE='OUTPUT.DAT')
c
c - Input the system matrices and program parameters
c
c     PRINT *,'CALLING INPDAT'
      CALL INPDAT
```

```
C
C - Input the initial guess for the compensator
C
C      PRINT *,'CALLING INITGS'
       CALL INITGS
C
C - Evaluate the derivatives of the Lagrangian
C
C      PRINT *,'CALLING EVDELF'
       CALL EVDELF
C
C - Initialize the vector S
C
C      PRINT *,'CALCULATING S'
       DO 10 I=1,NUMB
          SUM=0.0D0
          DO 20 J=1,NUMB
             SUM=SUM-H(I,J)*DELF(J)
  20      CONTINUE
          S(I)=SUM
  10   CONTINUE
C
C - Initialize ALPHA STAR
C
       ALSTAR = 1.0D-08
C
C - All inputs and initializations are complete.  Begin the
C   iterations.
C
       PRINT *,'Engage...'
       PRINT *
       PRINT *
C
C  >>>  This is the return point for the iteration  <<<
C
  30   CONTINUE
C
C - Update the counters
C
       ICOUNT=ICOUNT+1
       JCOUNT=JCOUNT+1
C
C - Calculate ALPHA STAR, the step size in the S direction which
C   minimizes the function
C
C      PRINT *,'CALLING FINDAL'
       CALL FINDAL(ALSTAR,FSTAR)
C
C - Save current derivatives into last-pass derivatives
C
       DO 40 I=1,NUMB
          DELOLF(I)=DELF(I)
  40   CONTINUE
C
C - Evaluate the derivatives of the Lagrangian
C
       CALL EVDELF
```

```
C
C - Update the variables H and S
C
      CALL UPDATH(ALSTAR)
C
C - Check for convergence of the solution
C
      CALL CKSTOP(ISTOP,FSTAR)
C
C - Write updates to user terminal every iteration
C
      WRITE(*,50) KSTATE,GAMMA,ALSTAR,FSTAR,
     +              ICOUNT,AMU,CHECKSTOP
      WRITE(2,50)KSTATE,GAMMA,ALSTAR,FSTAR,
     +              ICOUNT,AMU,CHECKSTOP
  50  FORMAT(/,I2,' State (gam=',F7.4,')',E19.10,F17.8,
     +'      -->>> ',I3,/,2X,'         (mu= ',F7.5,')',
     +'    CHECKSTOP:',E19.8)
C
C - Write data to output files every 10 iterations
C
      IF(JCOUNT.GE.10)THEN
         WRITE(9,*)
         WRITE(9,*) 'OUT OF FINDAL',ALSTAR,FSTAR,
     +              '  ITERATION',ICOUNT
         CALL EVALUF(FTEMP)
         WRITE(9,60) (DELF(II),II=1,NUMB)
   60    FORMAT(4E20.12)
         CALL WRITER(ICOUNT)
         JCOUNT=0
      END IF
C
C - Check for completion of the program (quit after 300 iterations
C   if no solution has been reached)
C
      IF(ICOUNT.LT.300.AND.ISTOP.NE.1) GOTO 30
C
C - Write final results to the output files
C
      WRITE(9,*)
      WRITE(9,*) 'FINAL VALUES OF THE DERIVATIVES:   '
      WRITE(9,60) (DELF(II),II=1,NUMB)
      CALL WRITER(ICOUNT)
      PRINT*,'^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^'
C
C - Clean up shop and go home
C
      CLOSE(1)
      CLOSE(2)
      CLOSE(8)
      CLOSE(9)
      STOP
      END
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                          c
c                        INPDAT                            c
c                                                          c
c  This subroutine reads the data for the system state-   c
c  space matrices and the program parameters from an      c
c  input file                                              c
c                                                          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE INPDAT
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*50 CHAR
c
c - INCLUDE the file containing all the array dimensions.
c   NOTE:  The file DIMEN.INC must be modified every time
c   there is a change in the size of the system or compensator,
c   and this subroutine must be re-compiled.
c
      INCLUDE 'DIMEN.INC'
c
c - State space system matrices
c
      COMMON/SYSTEM/
     .   A(ISTATE,ISTATE),
     .   BU(ISTATE,NUMBU), BD(ISTATE,NUMBD), BW(ISTATE,NUMBW),
     .   CY(NUMBY,ISTATE), CE(NUMBE,ISTATE), CZ(NUMBZ,ISTATE),
     .   DYD(NUMBY,NUMBD), DYW(NUMBY,NUMBW), DEU(NUMBE,NUMBU),
     .   DZU(NUMBZ,NUMBU)
c
c - Optimization parameters
c
      COMMON/PARAM/ AMU, OMAMU, GAMMA, GAM2INV
c
c - DFP algorithm variables
c
      COMMON/HMATRX/ H(NUMB,NUMB), S(NUMB), DELF(NUMB), DELOLF(NUMB)
      COMMON/MATRIX/ XMATOL(NUMB), TOLSER
      COMMON/FLAGS/  IFLAG2, ICNT, TOLCHK, CHECKSTOP
c
c - Format statements
c
 500  FORMAT(A50)
 510  FORMAT(8I7)
 520  FORMAT(2D11.3)
 530  FORMAT(1D11.3)
 540  FORMAT(8E15.5)
 541  FORMAT(8E15.5)
c
c - All inputs that are read from the input file are echoed
c   to the output files.
c
c - Input/output the title
c
      READ(1,500)  CHAR
      WRITE(9,500) CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(9,500) CHAR
      WRITE(8,500) CHAR
```

```
C
C - Input/output the dimensions of the matrices
C   NOTE: Dimensions are not really input here; they are in the
C         INCLUDE file.  They appear here for convenience.
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,510) ISTATE,KSTATE,NUMBU,NUMBY,NUMBD,NUMBE,NUMBW,NUMBZ
C
C - Input/output the parameters GAMMA and AMU
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,520)  GAMMA,AMU
      WRITE(8,520) GAMMA,AMU
      GAM2INV = 1.0D0/(GAMMA*GAMMA)
      OMAMU = 1.0D0 - AMU
C
C - Input/output the tolerance for the 1-D search and checkstop
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,520)  TOLSER,TOLCHK
      WRITE(8,520) TOLSER,TOLCHK
C
C - Input/output the system A matrix
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 10 I=1,ISTATE
         READ(1,540)  (A(I,J),J=1,ISTATE)
         WRITE(8,541) (A(I,J),J=1,ISTATE)
  10  CONTINUE
C
C - Input/output the system BU matrix
C   Note: Input file contains BU transpose
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 20 I=1,NUMBU
         READ(1,540)  (BU(J,I),J=1,ISTATE)
         WRITE(8,541) (BU(J,I),J=1,ISTATE)
  20  CONTINUE
C
C - Input/output the system BD matrix
C   Note: Input file contains BD transpose
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
```

```
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 30 I=1,NUMBD
         READ(1,540)  (BD(J,I),J=1,ISTATE)
         WRITE(8,541) (BD(J,I),J=1,ISTATE)
   30 CONTINUE
C
C - Input/output the system BW matrix
C   Note: Input file contains BW transpose
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 40 I=1,NUMBW
         READ(1,540)  (BW(J,I),J=1,ISTATE)
         WRITE(8,541) (BW(J,I),J=1,ISTATE)
   40 CONTINUE
C
C - Input/output the system CY matrix
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 50 I=1,NUMBY
         READ(1,540)  (CY(I,J),J=1,ISTATE)
         WRITE(8,541) (CY(I,J),J=1,ISTATE)
   50 CONTINUE
C
C - Input/output the system CE matrix
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 60 I=1,NUMBE
         READ(1,540)  (CE(I,J),J=1,ISTATE)
         WRITE(8,541) (CE(I,J),J=1,ISTATE)
   60 CONTINUE
C
C - Input/output the system CZ matrix
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 70 I=1,NUMBZ
         READ(1,540)  (CZ(I,J),J=1,ISTATE)
         WRITE(8,541) (CZ(I,J),J=1,ISTATE)
   70 CONTINUE
C
C - Input/output the system DYD matrix
C
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      READ(1,500)  CHAR
      WRITE(8,500) CHAR
      DO 80 I=1,NUMBY
         READ(1,540)  (DYD(I,J),J=1,NUMBD)
```

```
          WRITE(8,541) (DYD(I,J),J=1,NUMBD)
  80   CONTINUE
C
C - Input/output the system DYW matrix
C
       READ(1,500)  CHAR
       WRITE(8,500) CHAR
       READ(1,500)  CHAR
       WRITE(8,500) CHAR
       DO 90 I=1,NUMBY
          READ(1,540)  (DYW(I,J),J=1,NUMBW)
          WRITE(8,541) (DYW(I,J),J=1,NUMBW)
  90   CONTINUE
C
C - Input/output the system DEU matrix
C
       READ(1,500)  CHAR
       WRITE(8,500) CHAR
       READ(1,500)  CHAR
       WRITE(8,500) CHAR
       DO 100 I=1,NUMBE
          READ(1,540)  (DEU(I,J),J=1,NUMBU)
          WRITE(8,541) (DEU(I,J),J=1,NUMBU)
 100   CONTINUE
C
C - Input/output the system DZU matrix
C
       READ(1,500)  CHAR
       WRITE(8,500) CHAR
       READ(1,500)  CHAR
       WRITE(8,500) CHAR
       DO 110 I=1,NUMBZ
          READ(1,540)  (DZU(I,J),J=1,NUMBU)
          WRITE(8,541) (DZU(I,J),J=1,NUMBU)
 110   CONTINUE
C
C - Initialize the H matrix to the identity
C
       DO 120 I=1,NUMB
          DO 120 J=1,NUMB
             H(I,J)=0.0D0
             IF(I.EQ.J)H(I,J)=1.0D0
 120   CONTINUE
       RETURN
       END
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                       c
c                       INITGS                          c
c                                                       c
c    This subroutine reads the data for the initial guess c
c    of the state-space compensator matrices from an    c
c    input file                                         c
c                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE INITGS
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*50 CHAR
c
c - INCLUDE the file containing all the array dimensions.
c   NOTE:  The file DIMEN.INC must be modified every time
c   there is a change in the size of the system or compensator,
c   and this subroutine must be re-compiled.
c
      INCLUDE 'DIMEN.INC'
c
c - Compensator system matrices
c
      COMMON/COMP/ AC(KSTATE,KSTATE), BC(KSTATE,NUMBY),
     *             CC(NUMBU,KSTATE)
c
 500  FORMAT(A50)
 510  FORMAT(4D19.11)
c
c - Input the AC matrix
c
      IDONE=0
      J2 = 0
      DO 5 WHILE(IDONE.EQ.0)
         J1 = J2+1
         J2 = J1+3
         IF (J2.GE.KSTATE) THEN
            J2 = KSTATE
            IDONE = 1
         ENDIF
         READ(1,500)  CHAR
         WRITE(9,500) CHAR
         READ(1,500)  CHAR
         WRITE(9,500) CHAR
         DO 10 I=1,KSTATE
            READ(1,510)  (AC(I,J),J=J1,J2)
            WRITE(9,510) (AC(I,J),J=J1,J2)
  10     CONTINUE
   5  CONTINUE
c
c - Input the BC matrix   (NO TRANSPOSE)
c
      READ(1,500)  CHAR
      WRITE(9,500) CHAR
      READ(1,500)  CHAR
      WRITE(9,500) CHAR
      DO 20 I=1,KSTATE
         READ(1,510)  (BC(I,J),J=1,NUMBY)
         WRITE(9,510) (BC(I,J),J=1,NUMBY)
  20  CONTINUE
```

```
C
C - Input the CC matrix
C
      IDONE = 0
      J2 = 0
      DO 35 WHILE(IDONE.EQ.0)
         J1 = J2+1
         J2 = J1+3
         IF (J2.GE.KSTATE) THEN
            J2 = KSTATE
            IDONE = 1
         ENDIF
         READ(1,500)  CHAR
         WRITE(9,500) CHAR
         READ(1,500)  CHAR
         WRITE(9,500) CHAR
         DO 30 I=1,NUMBU
            READ(1,510)  (CC(I,J),J=J1,J2)
            WRITE(9,510) (CC(I,J),J=J1,J2)
 30      CONTINUE
 35   CONTINUE
      RETURN
      END
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                          c
c                        FINDAL                            c
c                                                          c
c   The subroutine calculates the ALPHA that minimizes     c
c   the function F(X + ALPHA*S)                            c
c                                                          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE FINDAL(ALSTAR,FSTAR)
      IMPLICIT REAL*8(A-H,O-Z)
c
c - INCLUDE the file containing all the array dimensions.
c   NOTE:  The file DIMEN.INC must be modified every time
c   there is a change in the size of the system or compensator,
c   and this subroutine must be re-compiled.
c
      INCLUDE 'DIMEN.INC'
c
c - DFP algorithm variables
c
      COMMON/MATRIX/ XMATOL(NUMB), TOLSER
      COMMON/HMATRX/ H(NUMB,NUMB), S(NUMB), DELF(NUMB), DELOLF(NUMB)
c
c - Compensator system matrices
c
      COMMON/COMP/ AC(KSTATE,KSTATE), BC(KSTATE,NUMBY),
     +             CC(NUMBU,KSTATE)
c
c
c - XMAT is a vector containing the matrices AC, BC, and CC
c
      DIMENSION XMAT(NUMB)
      EQUIVALENCE (XMAT(1),AC(1,1))
      EQUIVALENCE (XMAT(KSTATE*KSTATE+1),BC(1,1))
      EQUIVALENCE (XMAT(KSTATE*KSTATE+KSTATE*NUMBY+1),CC(1,1))
c
c - Save last-pass XMAT vector
c
      DO 10 I=1,NUMB
         XMATOL(I)=XMAT(I)
 10   CONTINUE
c
c - Initialize starting ALPHA to last-pass ALPHA STAR
c
      SHIFT=ALSTAR
c
c - Identify an initial region
c
      II = 0
      DO 30 I=1,3
c        PRINT *,'I = ',I
         ALPHA=(I-1)*SHIFT
         DO 40 J=1,NUMB
c           PRINT *,'XMATOL:',J,XMATOL(J)
            XMAT(J)=XMATOL(J)+ALPHA*S(J)
c        PRINT *,'XMAT(J) = ',J,XMAT(J)
 40      CONTINUE
         CALL EVALUF(FF)
c         PRINT *,'FUNCTION = ',FF
```

```
C
C - Identify three points in the region
C
        II = II + 1
        IF (II.EQ.1) THEN
           F0=FF
           F1 = FF
           ALPHA1 = ALPHA
C          PRINT *,'F1,ALPHA1:',F1,ALPHA1
        ELSE IF (II.EQ.2) THEN
           F2 = FF
           ALPHA2 = ALPHA
C          PRINT *,'F2,ALPHA2:',F2,ALPHA2
        ELSE
           F3 = FF
           ALPHA3 = ALPHA
C          PRINT *,'F3,ALPHA3:',F3,ALPHA3
        END IF
   30 CONTINUE
C
C - Expand upper bound until the minimizing ALPHA is within the
C   region
C
        DO 35 WHILE (F3.LT.F2)
           F2 = F3
           ALPHA2 = ALPHA3
           ALPHA3 = ALPHA3 * 2.0D00
           DO 36 I=1,NUMB
              XMAT(I) = XMATOL(I) + ALPHA3 * S(I)
 36        CONTINUE
           CALL EVALUF(F3)
 35     CONTINUE
C
C - Return point if the ALPHA needs refining
C
  50    CONTINUE
C       WRITE(2,1000) ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,
     +                 F1,F2,F3,F4,F5
C1000 FORMAT(5E16.5,/,5F16.10,/)
C
C - Determine which region contains the min
C
        IF(F1.LT.F2)THEN
C       WRITE(2,1001) ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,
     +                 F1,F2,F3,F4,F5
C1001 FORMAT('1 ',5E16.5,/,5F16.10,/)
C
C - The min definitely lies between ALPHA1 and ALPHA2, shrink the
C   search area
C        ALPHA1 ----HERE---- ALPHA2 ------------- ALPHA3
C
              ALPHA3=ALPHA2
              ALPHA2=(ALPHA3-ALPHA1)/2.0D0
              F3 = F2
              DO 60 I=1,NUMB
                 XMAT(I) = XMATOL(I) + ALPHA2 * S(I)
 60           CONTINUE
              CALL EVALUF(F2)
           ELSE
```

```
C     WRITE(2,1002) ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,
     +                 F1,F2,F3,F4,F5
C1002 FORMAT('2 ',5E16.5,/,5F16.10,/)
C
C - The min lies to the left or right of ALPHA2, determine which
C   side
C
            ALPHA4=(ALPHA2-ALPHA1)/2.0D0+ALPHA1
            DO 90 I=1,NUMB
               XMAT(I)=XMATOL(I)+ALPHA4*S(I)
  90        CONTINUE
            CALL EVALUF(F4)
            IF(F4.LT.F2)THEN
C
C - The min lies between ALPHA1 and ALPHA2
C        ALPHA1 ----HERE---- ALPHA2 ------------- ALPHA3
C
               ALPHA3=ALPHA2
               ALPHA2=ALPHA4
               F3 = F2
               F2 = F4
C     WRITE(2,1003) ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,
     +                 F1,F2,F3,F4,F5
C1003 FORMAT('3 ',5E16.5,/,5F16.10,/)
            ELSE
               ALPHA5=(ALPHA3-ALPHA2)/2.0D0+ALPHA2
               DO 100 I=1,NUMB
                  XMAT(I)=XMATOL(I)+ALPHA5*S(I)
 100           CONTINUE
               CALL EVALUF(F5)
               IF(F5.LT.F2)THEN
C
C - The min lies between ALPHA2 and ALPHA3
C         ALPHA1 ------------ ALPHA2 ----HERE----- ALPHA3
C
                  ALPHA1=ALPHA2
                  ALPHA2=ALPHA5
                  F1 = F2
                  F2 = F5
C     WRITE(2,1004) ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,
     +                 F1,F2,F3,F4,F5
C1004 FORMAT('4 ',5E16.5,/,5F16.10,/)
               ELSE
C
C - The min lies between ALPHA 4 and ALPHA 5
C        ALPHA1 ----ALPHA4--- HERE ---ALPHA5----- ALPHA3
C
                  ALPHA1=ALPHA4
                  ALPHA3=ALPHA5
                  F1 = F4
                  F3 = F5
C     WRITE(2,1005) ALPHA1,ALPHA2,ALPHA3,ALPHA4,ALPHA5,
     +                 F1,F2,F3,F4,F5
C1005 FORMAT('5 ',5E16.5,/,5F16.10,/)
               END IF
            END IF
         END IF
```

```
C
C - Check for converence on ALPHA
C
      IF(ABS((ALPHA2-ALPHA1)/ALPHA2).GT.TOLSER)GOTO 50
C
C - Update XMAT using new ALPHA STAR
C
      ALSTAR=ALPHA1
      IF(F2.LT.F1)THEN
          IF(F3.LT.F2)THEN
              ALSTAR=ALPHA3
          ELSE
              ALSTAR=ALPHA2
          ENDIF
      ELSE
          IF(F3.LT.F1)THEN
              ALSTAR=ALPHA3
          ENDIF
      ENDIF
      DO 110 I=1,NUMB
          XMAT(I)=XMATOL(I)+ALSTAR*S(I)
 110  CONTINUE
      CALL EVALUF(FSTAR)
      RETURN
      END
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                          c
c                         UPDATH                           c
c                                                          c
c  This subroutine updates the variables H and S           c
c                                                          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE UPDATH(ALSTAR)
      IMPLICIT REAL*8(A-H,O-Z)
c
c - INCLUDE the file containing all the array dimensions.
c   NOTE:  The file DIMEN.INC must be modified every time
c   there is a change in the size of the system or compensator,
c   and this subroutine must be re-complied.
c
      INCLUDE 'DIMEN.INC'
c
c
c - DFP algorithm variables
c
      COMMON/HMATRX/ H(NUMB,NUMB), S(NUMB), DELF(NUMB), DELOLF(NUMB)
      DIMENSION Z(NUMB), AM(NUMB,NUMB), AN(NUMB,NUMB), HY(NUMB)
c
c - Update Z
c
      DO 10 I=1,NUMB
         Z(I)=DELF(I)-DELOLF(I)
 10   CONTINUE
      TR2Z=0.0D0
      DO 11 I=1,NUMB
         TR2Z=Z(I)*Z(I)
 11   CONTINUE
      IF(TR2Z.EQ.0.0D0)THEN
    ISTOP=1
    RETURN
      END IF
c
c - Find the denominator for M
c
      SUM=0.0D0
      DO 20 I=1,NUMB
         SUM=SUM+S(I)*Z(I)
 20   CONTINUE
      FACM=ALSTAR/SUM
c
c - Store M
c
      DO 30 I=1,NUMB
         DO 30 J=1,NUMB
            AM(I,J)=S(I)*S(J)*FACM
 30   CONTINUE
c
c - Find the denominator for N
c
      SUM=0.0D0
      DO 40 I=1,NUMB
         DO 40 J=1,NUMB
            SUM=SUM+Z(I)*H(I,J)*Z(J)
 40   CONTINUE
      FACN=-SUM
```

```
C
C - Calculate N
C
      DO 50 I=1,NUMB
          SUM=0.0D0
          DO 60 J=1,NUMB
              SUM=SUM+H(I,J)*Z(J)
  60      CONTINUE
          HY(I)=SUM
  50  CONTINUE
      DO 70 I=1,NUMB
          DO 70 J=1,NUMB
              AN(I,J)=HY(I)*HY(J)/FACN
  70  CONTINUE
C
C - Update H
C
      DO 80 I=1,NUMB
          DO 80 J=1,NUMB
              H(I,J)=H(I,J)+AM(I,J)+AN(I,J)
  80  CONTINUE
C
C - Update S
C
      DO 90 I=1,NUMB
          SUM=0.0D0
          DO 100 J=1,NUMB
              SUM=SUM-H(I,J)*DELF(J)
 100      CONTINUE
          S(I)=SUM
  90  CONTINUE
C
C - Check to insure H direction will decrease function
C
      SUM=0.0D0
      DO 200 I=1,NUMB
        SUM=SUM+S(I)*DELF(I)
 200  CONTINUE
      IF(SUM.GT.0.0D0)THEN
       WRITE(*,*) 'HAD TO UPDATE H'
       WRITE(9,*) 'HAD TO UPDATE H'
       DO 210 I=1,NUMB
         DO 210 J=1,NUMB
             H(I,J)=0.0D0
             IF(I.EQ.J)H(I,J)=1.0D0
 210  CONTINUE
C
C - Update S
C
      DO 290 I=1,NUMB
        SUM=0.0D0
        DO 300 J=1,NUMB
            SUM=SUM-H(I,J)*DELF(J)
 300      CONTINUE
        S(I)=SUM
 290  CONTINUE
      END IF
      RETURN
      END
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                        c
c                       CKSTOP                           c
c                                                        c
c  This subroutine determines if the solution converged  c
c                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE CKSTOP(ISTOP,FSTAR)
      IMPLICIT REAL*8(A-H,O-Z)
c
c - INCLUDE the file containing all the array dimensions.
c   NOTE:  The file DIMEN.INC must be modified every time
c   there is a change in the size of the system or compensator,
c   and this subroutine must be re-compiled.
c
      INCLUDE 'DIMEN.INC'
c
c - DFP algorithm variables
c
      COMMON/HMATRX/ H(NUMB,NUMB), S(NUMB), DELF(NUMB), DELOLF(NUMB)
      COMMON/FLAGS/  IFLAG2, ICNT, TOLCHK, CHECKSTOP
c
c  - Set the STOP flag to zero
c
      ISTOP=0
      TR2Z=0.0D0
      DO 21 I=1,NUMB
         TR2Z=(DELF(I)-DELOLF(I))*(DELF(I)-DELOLF(I))
 21   CONTINUE
      IF(TR2Z.EQ.0.0D0)THEN
         ISTOP=1
         RETURN
      END IF
c
c - Calculate the magnitude of the residuals
c
      SUM=0.0D0
      DO 10 I=1,NUMB
         DO 10 J=1,NUMB
            SUM=SUM+DELF(I)*H(I,J)*DELF(J)
 10   CONTINUE
      CHECKSTOP = ABS(SUM/FSTAR)
      IF(CHECKSTOP.LT.TOLCHK)ISTOP=1
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                          C
C                        WRITER                            C
C                                                          C
C   This subroutine writes output date to the RESET file   C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE WRITER
      IMPLICIT REAL*8(A-H,O-Z)
C
C - INCLUDE the file containing all the array dimensions.
C   NOTE:  The file DIMEN.IND must be modified every time
C   there is a change in the size of the system or compensator,
C   and this subroutine must be re-complied.
C
      INCLUDE 'DIMEN.INC'
C
C - Compensator system matrices
C
      COMMON/COMP/ AC(KSTATE,KSTATE), BC(KSTATE,NUMBY),
     +              CC(NUMBU,KSTATE)
C
C - DFP algorithm variables
C
      COMMON/HMATRX/ H(NUMB,NUMB), S(NUMB), DELF(NUMB), DELOLF(NUMB)
C
C - Assorted and various variables
C
      COMMON/OUTDAT/ TRACE, TWONORM
C
 500  FORMAT(A50)
 501  FORMAT(/,'   THE AC MATRIX  (COLUMNS ',I3,' - ',I3,')')
 502  FORMAT(/,'   THE BC MATRIX')
 503  FORMAT(/,'   THE CC MATRIX  (COLUMNS ',I3,' - ',I3,')')
 510  FORMAT(4D19.11)
C
C - Output the AC matrix
C
      IDONE = 0
      J2 = 0
      DO 10 WHILE(IDONE.EQ.0)
         J1 = J2+1
         J2 = J1+3
         IF (J2.GE.KSTATE) THEN
            J2 = KSTATE
            IDONE = 1
         ENDIF
         WRITE(8,501) J1,J2
         DO 20 I=1,KSTATE
            WRITE(8,510) (AC(I,J),J=J1,J2)
  20     CONTINUE
  10  CONTINUE
C
C - Output the BC matrix  (NO TRANSPOSE)
C
      WRITE(8,502)
      DO 30 I=1,KSTATE
         WRITE(8,510) (BC(I,J),J=1,NUMBY)
  30  CONTINUE
```

```
C
C - Output the CC matrix
C
      IDONE = 0
      J2 = 0
      DO 40 WHILE(IDONE.EQ.0)
         J1 = J2+1
         J2 = J1+3
         IF (J2.GE.KSTATE) THEN
            J2 = KSTATE
            IDONE = 1
         ENDIF
         WRITE(8,503) J1,J2
         DO 50 I=1,NUMBU
            WRITE(8,510) (CC(I,J),J=J1,J2)
  50     CONTINUE
  40  CONTINUE
C
      WRITE(9,*) 'The two norm is',TWONORM
      WRITE(*,*) 'The two norm is',TWONORM
      RETURN
      END
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                        c
c                       EVALUF                           c
c                                                        c
c  This subroutine evaluates the value of the cost       c
c  function                                              c
c                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE EVALUF(FU)
      IMPLICIT REAL*8(A-H,O-Z)
c
c - INCLUDE the file containing all the array dimensions.
c   NOTE:  The file DIMEN.INC must be modified every time
c   there is a change in the size of the system or compensator,
c   and this subroutine must be re-compiled.
c
      INCLUDE 'DIMEN.INC'
c
c - State space system matrices
c
      COMMON/SYSTEM/
     .   A(ISTATE,ISTATE),
     .   BU(ISTATE,NUMBU), BD(ISTATE,NUMBD), BW(ISTATE,NUMBW),
     .   CY(NUMBY,ISTATE), CE(NUMBE,ISTATE), CZ(NUMBZ,ISTATE),
     .   DYD(NUMBY,NUMBD), DYW(NUMBY.NUMBW), DEU(NUMBE,NUMBU),
     .   DZU(NUMBZ,NUMBU)
c
c - Tilde matrices
c
      COMMON/QTWOTL/
     .   QTWO1(ISTATE,ISTATE),  QTWO12(ISTATE,KSTATE),
     .   QTWO21(KSTATE,ISTATE), QTWO2(KSTATE,KSTATE)
      COMMON/QINFTL/
     .   QINF1(ISTATE,ISTATE),  QINF12(ISTATE,KSTATE),
     .   QINF21(KSTATE,ISTATE), QINF2(KSTATE,KSTATE)
      COMMON/XTL/
     .   XTL1(ISTATE,ISTATE),  XTL12(ISTATE,KSTATE),
     .   XTL21(KSTATE,ISTATE), XTL2(KSTATE,KSTATE)
      COMMON/YTL/
     .   YTL1(ISTATE,ISTATE),  YTL12(ISTATE,KSTATE),
     .   YTL21(KSTATE,ISTATE), YTL2(KSTATE,KSTATE)
c
c - Compensator system matrices
c
      COMMON/COMP/ AC(KSTATE,KSTATE), BC(KSTATE,NUMBY),
     +                CC(NUMBU,KSTATE)
c
c - Optimization parameters
c
      COMMON/PARAM/ AMU, OMAMU, GAMMA, GAM2INV
c
c - DFP algorithm variables
c
      COMMON/MATRIX/ XMATOL(NUMB), TOLSER
c
c - Riccati solution matrices
c
      COMMON/RICINP/
     .   F(TILDIM,TILDIM), G(TILDIM,TILDIM), H(TILDIM,TILDIM),
     .   X(TILDIM,TILDIM)
```

```
      COMMON/RICSCR/
     .   Z(4*TILDIM*TILDIM), W(4*TILDIM*TILDIM),
     .   ER(2*TILDIM), EI(2*TILDIM),
     .   WORK(2*TILDIM), IND(2*TILDIM)
      COMMON/RSOLN/
     .   QTLINF(TILDIM,TILDIM), QTLTWO(TILDIM,TILDIM),
     .   XTILDE(TILDIM,TILDIM), YTILDE(TILDIM,TILDIM)
C
C - Assorted and various variables
C
      COMMON/OUTDAT/ TRACE, TWONORM
      COMMON/FLAGS/  IFLAG2, ICNT, TOLCHK, CHECKSTOP
C
      COMMON/TILDES/ RTLTWO(TILDIM,TILDIM), RTLINF(TILDIM,TILDIM),
     .   VTLTWO(TILDIM,TILDIM), VTLINF(TILDIM,TILDIM),
     .   ATIL(TILDIM,TILDIM),
BDTIL(TILDIM,TILDIM),BWTIL(TILDIM,TILDIM),
     .   CETIL(TILDIM,TILDIM), CZTIL(TILDIM,TILDIM)
C
C*****************************************************************
      N    = TILDIM
      NDIM = TILDIM
      ICNT = ICNT+1
C
C -  Calculate ATILDE
C        ATIL=  [A         BU*CC]
C               [BC*CY       AC]
C
      DO 40 I=1,ISTATE
      DO 40 J=1,ISTATE
         ATIL(I,J) = A(I,J)
  40  CONTINUE
      DO 41 I=1,KSTATE
      DO 41 J=1,KSTATE
         ATIL(I+ISTATE,J+ISTATE) = AC(I,J)
  41  CONTINUE
      DO 42 I=1,ISTATE
      DO 42 J=1,KSTATE
      SUM1 = 0.0D0
         DO 43 K=1,NUMBU
            SUM1 = SUM1+BU(I,K)*CC(K,J)
  43     CONTINUE
         ATIL(I,J+ISTATE) = SUM1
  42  CONTINUE
      DO 44 I=1,KSTATE
      DO 44 J=1,ISTATE
      SUM1 = 0.0D0
         DO 45 K=1,NUMBY
            SUM1 = SUM1+BC(I,K)*CY(K,J)
  45     CONTINUE
         ATIL(I+ISTATE,J) = SUM1
  44  CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)
C     WRITE(2,*)'ATIL'
C     DO 46 I=1,TILDIM
C        WRITE(2,1000) (ATIL(I,J),J=1,TILDIM)
C 46  CONTINUE
 1000 FORMAT(10E15.4)
```

```
C
C - Calculate the B TILDES
C
C        BDTIL = [   BD   ]        BWTIL = [   BW   ]
C                [BC*DYD]                  [BC*DYW]
C
      DO 70 I=1,ISTATE
      DO 70 J=1,NUMBD
         BDTIL(I,J)=BD(I,J)
  70  CONTINUE
      DO 72 I=1,ISTATE
      DO 72 J=1,NUMBW
         BWTIL(I,J)=BW(I,J)
  72  CONTINUE
      DO 73 I=1,KSTATE
         DO 74 J=1,NUMBD
            SUM1 = 0.0D0
            DO 75 K=1,NUMBY
               SUM1 = SUM1+BC(I,K)*DYD(K,J)
  75        CONTINUE
            BDTIL(I+ISTATE,J) = SUM1
  74     CONTINUE
         DO 76 J=1,NUMBW
            SUM1 = 0.0D0
            DO 77 K=1,NUMBY
               SUM1 = SUM1+BC(I,K)*DYW(K,J)
  77        CONTINUE
            BWTIL(I+ISTATE,J) = SUM1
  76     CONTINUE
  73  CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'BWTIL'
C     DO 78 I=1,TILDIM
C        WRITE(2,1000) (BWTIL(I,J),J=1,NUMBW)
C 78  CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'BDTIL'
C     DO 79 I=1,TILDIM
C        WRITE(2,1000) (BDTIL(I,J),J=1,NUMBD)
C 79  CONTINUE
C
C - Calculate the C TILDES
C
C        CETIL = [CE  DEU*CC]      CZTIL = [CZ  DZU*CC]
C
      DO 80 I=1,ISTATE
         DO 81 J=1,NUMBE
            CETIL(J,I) = CE(J,I)
  81     CONTINUE
         DO 82 J=1,NUMBZ
            CZTIL(J,I) = CZ(J,I)
  82     CONTINUE
  80  CONTINUE
      DO 83 I=1,KSTATE
         DO 84 J=1,NUMBE
            SUM1 = 0.0D0
            DO 85 K=1,NUMBU
               SUM1= SUM1+DEU(J,K)*CC(K,I)
  85        CONTINUE
            CETIL(J,I+ISTATE) = SUM1
```

```
 84      CONTINUE
         DO 86 J=1,NUMBZ
            SUM1 = 0.0D0
            DO 87 K=1,NUMBU
               SUM1 = SUM1+DZU(J,K)*CC(K,I)
 87         CONTINUE
            CZTIL(J,I+ISTATE) = SUM1
 86      CONTINUE
 83   CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'CZTIL'
C     DO 88 I=1,NUMBZ
C        WRITE(2,1000) (CZTIL(I,J),J=1,TILDIM)
C 88  CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'CETIL'
C     DO 89 I=1,NUMBE
C        WRITE(2,1000) (CETIL(I,J),J=1,TILDIM)
C 89  CONTINUE
C
C - Calculate the R TILDES
C
C        RTLTWO=CZ TILDE' * CZ TILDE
C        RTLINF=CE TILDE' * CE TILDE
C
      DO 160 I=1,TILDIM
      DO 160 J=1,TILDIM
         SUM1=0.0D0
         DO 170 K=1,NUMBZ
            SUM1=SUM1+CZTIL(K,I)*CZTIL(K,J)
 170        CONTINUE
         RTLTWO(I,J)=SUM1
         SUM1=0.0D0
         DO 180 K=1,NUMBE
            SUM1=SUM1+CETIL(K,I)*CETIL(K,J)
 180        CONTINUE
         RTLINF(I,J)=SUM1
 160    CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'RTLTWO'
C     DO 181 I=1,TILDIM
C        WRITE(2,1000) (RTLTWO(I,J),J=1,TILDIM)
C 181 CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'RTLINF'
C     DO 182 I=1,TILDIM
C        WRITE(2,1000) (RTLINF(I,J),J=1,TILDIM)
C 182 CONTINUE
C
C - Calculate the V TILDES
C
C        VTLTWO=BW TILDE * BW TILDE'
C        VTLINF=BD TILDE * BD TILDE'
C
      DO 190 I=1,TILDIM
      DO 190 J=1,TILDIM
         SUM1=0.0D0
         DO 200 K=1,NUMBW
            SUM1=SUM1+BWTIL(I,K)*BWTIL(J,K)
 200        CONTINUE
```

```
          VTLTWO(I,J)=SUM1
          SUM1=0.0D0
          DO 210 K=1,NUMBD
             SUM1=SUM1+BDTIL(I,K)*BDTIL(J,K)
 210         CONTINUE
          VTLINF(I,J)=SUM1
 190    CONTINUE
C      WRITE(2,*)
C      WRITE(2,*)'VTLTWO'
C      DO 211 I=1,TILDIM
C         WRITE(2,1000) (VTLTWO(I,J),J=1,TILDIM)
C 211 CONTINUE
C      WRITE(2,*)
C      WRITE(2,*)'VTLINF'
C      DO 212 I=1,TILDIM
C         WRITE(2,1000) (VTLINF(I,J),J=1,TILDIM)
C 212 CONTINUE
C ------------------------------------------------------------
C
C  - Use Riccati solver
C       (solving F'X + XF - XGX + H = 0  FOR X)
       IFAIL=0
C
C       Q2 SOLUTION
C
C   F'X     +       XF      -      XGX      +      H       =0
C  ATILDE*Q2 + Q2*ATILDE'  - X*0*X     +   BW*BW'     =0
C    thus  F=ATILDE'
C          G=0
C          H=BW*BW'
C
C NOTE: must "start" RICSOL with X=H
C
       DO 220 I=1,TILDIM
          DO 220 J=1,TILDIM
             F(I,J)=ATIL(J,I)
             G(I,J)=0.0D0
             X(I,J)=VTLTWO(I,J)
 220   CONTINUE
C
C - Call Riccati solver for solution to Q2
C
C      PRINT *,'CALLING RICSOL 1'
       CALL RICCATI SOLVER (output: X)
C      PRINT *,'BACK FROM RICSOL 1'
C
       DO 230 I=1,TILDIM
          DO 230 J=1,TILDIM
             QTLTWO(I,J)=X(I,J)
 230   CONTINUE
C      WRITE(2,*)
C      WRITE(2,*)'QTLTWO'
C      DO 231 I=1,TILDIM
C         WRITE(2,1000) (QTLTWO(I,J),J=1,TILDIM)
C 231 CONTINUE
C
C - Check for stable and unique solution
C
       CALL EIGENVALUE SOLVER  (output:  ER-real part of eig's)
```

```
      DO 301 I=1,N
C        WRITE(2,*) 'Q2 EIGENV-REAL (EIGINDEX)=',I,'EIG=',ER(I)
         IF(ER(I).LE.-1E-20) THEN
C           PRINT*,'Q2 PROBLEM (EIGINDEX)=',I,'EIG=',ER(I)
            IFAIL=1
            GO TO 306
         ENDIF
 301  CONTINUE
C
C------------------------------------------------------------------
C    QINF SOLUTION
C
C   F'X     +       XF      -      XGX      +      H       =0
C  ATILDE*QINF + QINF*ATILDE' + QINF*GAM2INV*CE'*CE*QINF + BD*BD'=0
C    thus  F=ATILDE'
C          G=-GAM2INV*CE'*CE
C          H=BD*BD'
C
C  NOTE: F is the same as for Q2
      DO 240 I=1,TILDIM
         DO 240 J=1,TILDIM
            G(I,J)=-GAM2INV * RTLINF(I,J)
            X(I,J)=VTLINF(I,J)
 240  CONTINUE
C
C - Call Riccati solver for solution to QINF
C
C     PRINT *,'CALLING RICSOL 2'
      CALL RICCATI SOLVER (output: X)
C     PRINT *,'BACK FROM RICSOL 2'
C
      DO 250 I=1,TILDIM
         DO 250 J=1,TILDIM
            QTLINF(I,J)=X(I,J)
 250  CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'QTLINF'
C     DO 251 I=1,TILDIM
C        WRITE(2,1000) (QTLINF(I,J),J=1,TILDIM)
C251  CONTINUE
C
C - Check for stable and unique solution
C
      CALL EIGENVALUE SOLVER  (output:  ER-real part of eig's)
      DO 302 I=1,N
C        WRITE(2,*) 'QINF EIGENV-REAL (EIGINDEX)=',I,'EIG=',ER(I)
         IF(ER(I).LE.-1E-20) THEN
C           PRINT*,'QINF PROBLEM (EIGINDEX)=',I,'EIG=',ER(I)
            IFAIL=1
            GO TO 306
         ENDIF
 302  CONTINUE
```

```
C
C-----------------------------------------------------------------
C    X LAGRANGE MULTIPLIER SOLUTION
C
C    F'X     +        XF       -       XGX       +       H        =0
C  ATILDE'*X + X*ATILDE    - X*O*X + (1-AMU)*CZ'*CZ =0
C    thus     F=ATILDE
C             G=0
C             H=(1-AMU)*CZ'*CZ
C
C        IF(IFLAG2.EQ.1) THEN
         DO 260 I=1,TILDIM
            DO 260 J=1,TILDIM
               F(I,J)=ATIL(I,J)
               G(I,J)=0.0D0
               X(I,J)=OMAMU*RTLTWO(I,J)
 260     CONTINUE
C
C - Call Riccati solver for solution of XTILDE
C
C        PRINT *,'CALLING RICSOL 3'
         CALL RICCATI SOLVER (output: X)
C        PRINT *,'BACK FROM RICSOL 3'
C
         DO 270 I=1,TILDIM
            DO 270 J=1,TILDIM
               XTILDE(I,J)=X(I,J)
 270     CONTINUE
C        WRITE(2,*)
C        WRITE(2,*)'XTILDE'
C        DO 271 I=1,TILDIM
C           WRITE(2,1000) (XTILDE(I,J),J=1,TILDIM)
C 271    CONTINUE
C        END IF
C
C - Check for stable and unique solution
C
         CALL EIGENVALUE SOLVER  (output:  ER-real part of eig's)
         DO 303 I=1,N
C           WRITE(2,*) 'XTILDE EIGENV-REAL (EIGINDEX)=',I,'EIG=',ER(I)
            IF(ER(I).LE.-1E-10) THEN
C              PRINT*,'XTILDE PROBLEM (EIGINDEX)=',I,'EIG=',ER(I)
               IFAIL=1
               GO TO 306
            ENDIF
 303     CONTINUE
C
C-----------------------------------------------------------------
-
C    Y LAGRANGE MULTIPLIER SOLUTION
C
C    F'X     +        XF       -       XGX       +       H        =0
C    P'*Y + Y*P - X*O*X + AMU*CE'*CE =0
C    where P=ATILDE+(GAM2INV*QINF*CE'*CE)
C      thus   F=P
C             G=0
C             H=AMU*CE'*CE
C
```

```
      DO 280 I=1,TILDIM
         DO 280 J=1,TILDIM
            SUM1=0.0D0
            G(I,J)=0.0D0
            X(I,J)=AMU*RTLINF(I,J)
            DO 290 K=1,TILDIM
               SUM1=SUM1 + QTLINF(I,K)*RTLINF(K,J)
 290        CONTINUE
            SUM1=SUM1*GAM2INV + ATIL(I,J)
            F(I,J)=SUM1
 280  CONTINUE
C
C - Call Riccati solver for solution to YTILDE
C
C     PRINT *,'CALLING RICSOL 4'
      CALL RICCATI SOLVER (output: X)
C     PRINT *,'BACK FROM RICSOL 4'
C
      DO 300 I=1,TILDIM
         DO 300 J=1,TILDIM
            YTILDE(I,J)=X(I,J)
 300  CONTINUE
C     WRITE(2,*)
C     WRITE(2,*)'YTILDE'
C     DO 299 I=1,TILDIM
C        WRITE(2,1000) (YTILDE(I,J),J=1,TILDIM)
C 299 CONTINUE
C
C - Check for stable and unique solution
C
      CALL EIGENVALUE SOLVER  (output:  ER-real part of eig's)
      DO 304 I=1,N
C        WRITE(2,*) 'YTILDE EIGENV-REAL (EIGINDEX)=',I,'EIG=',ER(I)
         IF(ER(I).LE.-1E-10) THEN
C           PRINT*,'YTILDE PROBLEM (EIGINDEX)=',I,'EIG=',ER(I)
            IFAIL=1
            GO TO 306
         ENDIF
 304  CONTINUE
      CALL EIGENVALUE SOLVER  (output:  ER-real part of eig's)
      DO 305 I=1,N
C        WRITE(2,*) 'Ast EIGENV-REAL (EIGINDEX)=',I,'EIG=',ER(I)
         IF(ER(I).GE.-1E-8) THEN
C           PRINT*,'Ast PROBLEM (EIGINDEX)=',I,'EIG=',ER(I)
            IFAIL=1
            GO TO 306
         ENDIF
 305  CONTINUE
C
C-----------------------------------------------------------------
C - If acceptable solutions were found to the Riccati and Lyapunov
C   equations, use these solutions to calculate the function value.
C   Otherwise, no acceptable solution exists.  Set the function
C   value to "very big."
C
  306 IF(IFAIL.EQ.1)THEN
         FU=1E+16
      ELSE
C
```

```
C
C - Calculate the cost function
C
C      PRINT *,'CALCULATING THE COST FUNCTION'
       TRACE=0.0D0
       SUM=0.0D0
       DO 390 I=1,TILDIM
          DO 390 J=1,TILDIM
             SUM=SUM+(1.0D0-AMU)*QTLTWO(I,J)*RTLTWO(J,I)
     .                  +AMU*QTLINF(I,J)*RTLINF(J,I)
 390   CONTINUE
       FU=TRACE+SUM
C      WRITE(2,*)
C      WRITE(2,1001) FU
 1001 FORMAT(F10.5)
C
C - Calculate the 2-norm of Tzw
C
       SUM=0.0D0
       DO 400 I=1,TILDIM
          DO 400 J=1,TILDIM
             SUM=SUM+QTLTWO(I,J)*RTLTWO(J,I)
 400      CONTINUE
          TWONORM = DSQRT(SUM)
       END IF
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                    C
C                       EVDELF                       C
C                                                    C
C This subroutine evaluates derivatives of the Laplacian  C
C with respect to the variable matrices AC, BC, and CC    C
C                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE EVDELF
      IMPLICIT REAL*8(A-H,O-Z)
C
C - INCLUDE the file containing all the array dimensions.
C   NOTE:  The file DIMEN.INC must be modified every time
C   there is a change in the size of the system or compensator,
C   and this subroutine must be re-compiled.
C
      INCLUDE 'DIMEN.INC'
      DIMENSION ACDER(KSTATE,KSTATE), BCDER(KSTATE,NUMBY),
     .          CCDER(NUMBU,KSTATE)
C
C - State space system matrices
C
      COMMON/SYSTEM/
     .   A(ISTATE,ISTATE),
     .   BU(ISTATE,NUMBU), BD(ISTATE,NUMBD), BW(ISTATE,NUMBW),
     .   CY(NUMBY,ISTATE), CE(NUMBE,ISTATE), CZ(NUMBZ,ISTATE),
     .   DYD(NUMBY,NUMBD), DYW(NUMBY,NUMBW), DEU(NUMBE,NUMBU),
     .   DZU(NUMBZ,NUMBU)
C
C - Tilde matrices
C
      COMMON/QTWOTL/
     .   QTWO1(ISTATE,ISTATE),   QTWO12(ISTATE,KSTATE),
     .   QTWO21(KSTATE,ISTATE), QTWO2(KSTATE,KSTATE)
      COMMON/QINFTL/
     .   QINF1(ISTATE,ISTATE),   QINF12(ISTATE,KSTATE),
     .   QINF21(KSTATE,ISTATE), QINF2(KSTATE,KSTATE)
      COMMON/XTL/
     .   XTL1(ISTATE,ISTATE),   XTL12(ISTATE,KSTATE),
     .   XTL21(KSTATE,ISTATE), XTL2(KSTATE,KSTATE)
      COMMON/YTL/
     .   YTL1(ISTATE,ISTATE),   YTL12(ISTATE,KSTATE),
     .   YTL21(KSTATE,ISTATE), YTL2(KSTATE,KSTATE)
C
C - Compensator system matrices
C
      COMMON/COMP/ AC(KSTATE,KSTATE), BC(KSTATE,NUMBY),
     +              CC(NUMBU,KSTATE)
C
C - Optimization parameters
C
      COMMON/PARAM/ AMU, OMAMU, GAMMA, GAM2INV
C
C - DFP algorithm variables
C
      COMMON/HMATRX/ H(NUMB,NUMB), S(NUMB), DELF(NUMB), DELOLF(NUMB)
      EQUIVALENCE (DELF(1),ACDER(1,1))
      EQUIVALENCE (DELF(KSTATE*KSTATE+1),BCDER(1,1))
      EQUIVALENCE (DELF(KSTATE*KSTATE+KSTATE*NUMBY+1),CCDER(1,1))
```

```
C
C - Riccati solution matrices
C
      COMMON/RSOLN/
     .   QTLINF(TILDIM,TILDIM), QTLTWO(TILDIM,TILDIM),
     .   XTILDE(TILDIM,TILDIM), YTILDE(TILDIM,TILDIM)
C
C - Assorted and various variables
C
      COMMON/FLAGS/  IFLAG2, ICNT, TOLCHK, CHECKSTOP
C-----------------------------------------------------------------
      IFLAG2 = 1
C     PRINT *,'CALLING EVALUF FROM EVDELF'
      CALL EVALUF(FU)
C     PRINT *,'BACK FROM EVALUF'
      IFLAG = 0
C
C - Partition the tilde matrices (XTILDE, YTILDE, QTLTWO, QTLINF)
C
C   -  Partition the 1,1 components
C
      DO 10 I=1,ISTATE
      DO 10 J=1,ISTATE
         XTL1(I,J)   = XTILDE(I,J)
         YTL1(I,J)   = YTILDE(I,J)
         QTWO1(I,J)  = QTLTWO(I,J)
         QINF1(I,J)  = QTLINF(I,J)
 10   CONTINUE
C
C   -  Partition the 1,2 components
C
      DO 11 I=1,ISTATE
      DO 11 J=1,KSTATE
         XTL12(I,J)   = XTILDE(I,J+ISTATE)
         YTL12(I,J)   = YTILDE(I,J+ISTATE)
         QTWO12(I,J)  = QTLTWO(I,J+ISTATE)
         QINF12(I,J)  = QTLINF(I,J+ISTATE)
 11   CONTINUE
C
C   -  Partition the     components
C
      DO 12 I=1,KST
      DO 12 J=1,IST
         XTL21(I,J)   - XTILDE(I+ISTATE,J)
         YTL21(I,J)   = YTILDE(I+ISTATE,J)
         QTWO21(I,J)  = QTLTWO(I+ISTATE,J)
         QINF21(I,J)  = QTLINF(I+ISTATE,J)
 12   CONTINUE
C
C   -  Partition the 2,2 components
C
      DO 13 I=1,KSTATE
      DO 13 J=1,KSTATE
         XTL2(I,J)   = XTILDE(I+ISTATE,J+ISTATE)
         YTL2(I,J)   = YTILDE(I+ISTATE,J+ISTATE)
         QTWO2(I,J)  = QTLTWO(I+ISTATE,J+ISTATE)
         QINF2(I,J)  = QTLINF(I+ISTATE,J+ISTATE)
 13   CONTINUE
```

```
C
C - Find the derivative wrt AC
C
      ICOUNT=0
      DO 20 I=1,KSTATE
          DO 20 J=1,KSTATE
              ICOUNT=ICOUNT+1
              SUM=0.0D0
              DO 25 K=1,ISTATE
                  SUM=SUM+XTL12(K,I)*QTWO21(J,K)
                  SUM=SUM+XTL21(I,K)*QTWO12(K,J)
                  SUM=SUM+YTL12(K,I)*QINF21(J,K)
                  SUM=SUM+YTL21(I,K)*QINF12(K,J)
 25           CONTINUE
              DO 30 K=1,KSTATE
                  SUM=SUM+XTL2(K,I)*QTWO2(J,K)
                  SUM=SUM+XTL2(I,K)*QTWO2(K,J)
                  SUM=SUM+YTL2(I,K)*QINF2(K,J)
                  SUM=SUM+YTL2(K,I)*QINF2(J,K)
 30           CONTINUE
          ACDER(I,J)=SUM
 20   CONTINUE
C
C - Find the derivative wrt BC
C
      DO 40 I=1,KSTATE
          DO 40 J=1,NUMBY
              ICOUNT=ICOUNT+1
              SUM=0.0D0
              DO 45 K=1,ISTATE
              DO 45 L=1,ISTATE
                  SUM=SUM+XTL12(K,I)*QTWO1(L,K)*CY(J,L)
                  SUM=SUM+XTL21(I,K)*QTWO1(K,L)*CY(J,L)
                  SUM=SUM+YTL12(K,I)*QINF1(L,K)*CY(J,L)
                  SUM=SUM+YTL21(I,K)*QINF1(K,L)*CY(J,L)
 45           CONTINUE
              DO 50 K=1,KSTATE
              DO 50 L=1,ISTATE
                  SUM=SUM+XTL2(I,K)*QTWO21(K,L)*CY(J,L)
                  SUM=SUM+XTL2(K,I)*QTWO12(L,K)*CY(J,L)
                  SUM=SUM+YTL2(I,K)*QINF21(K,L)*CY(J,L)
                  SUM=SUM+YTL2(K,I)*QINF12(L,K)*CY(J,L)
 50           CONTINUE
              DO 60 K=1,ISTATE
              DO 60 L=1,NUMBW
                  SUM=SUM+XTL12(K,I)*BW(K,L)*DYW(J,L)
                  SUM=SUM+XTL21(I,K)*DYW(J,L)*BW(K,L)
 60           CONTINUE
              DO 70 K=1,ISTATE
              DO 70 L=1,NUMBD
                  SUM=SUM+YTL12(K,I)*BD(K,L)*DYD(J,L)
                  SUM=SUM+YTL21(I,K)*DYD(J,L)*BD(K,L)
 70           CONTINUE
              DO 80 K=1,KSTATE
              DO 80 L=1,NUMBY
              DO 80 M=1,NUMBW
                  SUM=SUM+XTL2(I,K)*BC(K,L)*DYW(L,M)*DYW(J,M)
                  SUM=SUM+XTL2(K,I)*BC(K,L)*DYW(L,M)*DYW(J,M)
 80           CONTINUE
```

```
            DO 90 K=1,KSTATE
            DO 90 L=1,NUMBY
            DO 90 M=1,NUMBD
               SUM=SUM+YTL2(I,K)*BC(K,L)*DYD(L,M)*DYD(J,M)
               SUM=SUM+YTL2(K,I)*BC(K,L)*DYD(L,M)*DYD(J,M)
 90         CONTINUE
         BCDER(I,J)=SUM
 40   CONTINUE
C
C - Find the derivative wrt CC
C
      DO 100 I=1,NUMBU
      DO 100 J=1,KSTATE
         ICOUNT=ICOUNT+1
         SUM=0.0D0
         DO 110 K=1,ISTATE
         DO 110 L=1,ISTATE
            SUM=SUM+BU(K,I)*XTL1(L,K)*QTWO21(J,L)
            SUM=SUM+BU(K,I)*XTL1(K,L)*QTWO12(L,J)
            SUM=SUM+BU(K,I)*YTL1(L,K)*QINF21(J,L)
            SUM=SUM+BU(K,I)*YTL1(K,L)*QINF12(L,J)
 110     CONTINUE
         DO 111 K=1,ISTATE
         DO 111 L=1,KSTATE
            SUM=SUM+BU(K,I)*XTL12(K,L)*QTWO2(L,J)
            SUM=SUM+BU(K,I)*XTL21(L,K)*QTWO2(J,L)
            SUM=SUM+BU(K,I)*YTL12(K,L)*QINF2(L,J)
            SUM=SUM+BU(K,I)*YTL21(L,K)*QINF2(J,L)
 111     CONTINUE
         DO 120 K=1,NUMBE
         DO 120 L=1,ISTATE
            DO 121 M=1,ISTATE
            DO 121 N=1,ISTATE
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF1(L,M)*YTL1(M,N)*QINF12(N,J)
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF1(M,L)*YTL1(N,M)*QINF21(J,N)
 121        CONTINUE
            DO 122 M=1,ISTATE
            DO 122 N=1,KSTATE
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF1(M,L)*YTL21(N,M)*QINF2(J,N)
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF1(L,M)*YTL12(M,N)*QINF2(N,J)
 122        CONTINUE
            DO 123 M=1,KSTATE
            DO 123 N=1,ISTATE
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF21(M,L)*YTL12(N,M)*QINF21(J,N)
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF12(L,M)*YTL21(M,N)*QINF12(N,J)
 123        CONTINUE
            DO 124 M=1,KSTATE
            DO 124 N=1,KSTATE
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF12(L,M)*YTL2(M,N)*QINF2(N,J)
              SUM=SUM+GAM2INV*
     .         DEU(K,I)*CE(K,L)*QINF21(M,L)*YTL2(N,M)*QINF2(J,N)
 124        CONTINUE
 120     CONTINUE
```

```
         DO 130 K=1,NUMBE
         DO 130 L=1,NUMBU
         DO 130 M=1,KSTATE
           DO 131 N=1,ISTATE
           DO 131 O=1,ISTATE
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF21(M,N)*YTL1(N,O)*QINF12(O,J)
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF12(N,M)*YTL1(O,N)*QINF21(J,O)
 131       CONTINUE
           DO 132 N=1,ISTATE
           DO 132 O=1,KSTATE
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF21(M,N)*YTL12(N,O)*QINF2(O,J)
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF12(N,M)*YTL21(O,N)*QINF2(J,O)
 132       CONTINUE
           DO 133 N=1,KSTATE
           DO 133 O=1,ISTATE
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF2(M,N)*YTL21(N,O)*QINF12(O,J)
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF2(N,M)*YTL12(O,N)*QINF21(J,O)
 133       CONTINUE
           DO 134 N=1,KSTATE
           DO 134 O=1,KSTATE
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF2(M,N)*YTL2(N,O)*QINF2(O,J)
             SUM=SUM+GAM2INV*DEU(K,I)*DEU(K,L)*
     .        CC(L,M)*QINF2(N,M)*YTL2(O,N)*QINF2(J,O)
134        CONTINUE
130        CONTINUE
          DO 140 K=1,NUMBZ
          DO 140 L=1,ISTATE
                SUM=SUM+(1.0D0-AMU)*DZU(K,I)*CZ(K,L)*QTWO12(L,J)
                SUM=SUM+(1.0D0-AMU)*DZU(K,I)*CZ(K,L)*QTWO21(J,L)
140          CONTINUE
             DO 150 K=1,NUMBZ
             DO 150 L=1,NUMBU
             DO 150 M=1,KSTATE
                SUM=SUM+(1.0D0-AMU)
     .               *DZU(K,I)*DZU(K,L)*CC(L,M)*QTWO2(J,M)
                SUM=SUM+(1.0D0-AMU)
     .               *DZU(K,I)*DZU(K,L)*CC(L,M)*QTWO2(M,J)
150          CONTINUE
          DO 160 K=1,NUMBE
          DO 160 L=1,ISTATE
                SUM=SUM+AMU*DEU(K,I)*CE(K,L)*QINF12(L,J)
                SUM=SUM+AMU*DEU(K,I)*CE(K,L)*QINF21(J,L)
160          CONTINUE
             DO 170 K=1,NUMBE
             DO 170 L=1,NUMBU
             DO 170 M=1,KSTATE
                SUM=SUM+AMU*DEU(K,I)*DEU(K,L)*CC(L,M)*QINF2(J,M)
                SUM=SUM+AMU*DEU(K,I)*DEU(K,L)*CC(L,M)*QINF2(M,J)
170          CONTINUE
          CCDER(I,J)=SUM
100   CONTINUE
      RETURN
      END
```

```
THIS IS THE INPUT FILE FOR THE DIRECT METHOD
THE SISO MIX 8 STATE COMPENSATOR GAMMA 2.5

THE DIMENSIONS ISTATE,KSTATE,NU,NY,ND,NE,NW,NZ
      3       8       1       1       1       1       1       1

THE PARAMETERS GAMMA AND MU (2D11.6)
  0.250D+01  0.100D-00

THE TOLERANCES OF: 1-D SEARCH, CHECKSTOP (2D11.6)
  0.100D-03  0.100D-07

THE A MATRIX (8F8.4)
   -0.39080E+00   -0.45650E+00    0.12657E+01
    0.14453E+01   -0.10491E+01   -0.12077E+01
   -0.12880E+00    0.67440E+00    0.10324E+01

THE BU MATRIX AS BU TRANSPOSE
   -0.42750E+00   -0.44700E+00   -0.91720E+00

THE BD MATRIX AS BD TRANSPOSE
    0.48800E-01    0.36080E+00    0.35640E+00

THE BW MATRIX AS BW TRANSPOSE
    0.14077E+01    0.97230E+00   -0.16050E+01

THE CY MATRIX
   -0.15567E+01   -0.19432E+01   -0.91400E-01

THE CE MATRIX
    0.94200E+00    0.14400E-01    0.11870E+00

THE CZ MATRIX
   -0.45000E-01    0.36060E+00    0.18972E+01

THE DYD MATRIX
    0.51850E+00

THE DYW MATRIX
    0.38990E+00

THE DEU MATRIX
    0.13575E+01

THE DZU MATRIX
    0.57810E+00

   THE AC MATRIX  (COLUMNS    1 -    4)
 -0.42464048036D+01 -0.41938655996D+01 -0.20982704551D+01  0.38597033962D+01
  0.19735258928D+00 -0.13358453000D+01 -0.48825701128D+01  0.44280031937D+01
 -0.67612139041D+01 -0.52273410600D+01 -0.62417707662D+01  0.84680779691D+01
  0.93570728936D+01  0.84381238744D+01  0.76522396137D+01 -0.22312115910D+02
  0.34126422373D+01  0.30774899656D+01  0.27908680859D+01 -0.69399026978D+01
  0.40025571618D+01  0.36094699197D+01  0.32733021125D+01 -0.92716881072D+00
  0.56863512443D+00  0.51279002246D+00  0.46503134842D+00 -0.20016352363D+02
 -0.17570291239D+01 -0.15844730042D+01 -0.14369031873D+01  0.91291800491D+01

   THE AC MATRIX  (COLUMNS    5 -    8)
  0.36629700415D+01 -0.76222707011D+00 -0.27165890667D+01 -0.19927768328D+00
  0.42023029687D+01 -0.87445680506D+00 -0.31165775779D+01 -0.22861917803D+00
```

```
 0.80364506611D+01 -0.16723042154D+01 -0.59601180882D+01 -0.43720949159D+00
-0.13945635379D+02  0.51341155156D+01  0.84823843944D+01 -0.26400498205D+01
-0.67197475912D+01  0.95634835361D+00  0.91188220025D+01  0.13756622861D+01
-0.35055010102D+01  0.11063088353D+01  0.93504776975D+01  0.41454433150D+01
-0.10543648466D+02  0.20770424268D+01 -0.11037259724D+02 -0.75826660426D+01
 0.32035395034D+01 -0.34512450454D+01  0.31119395199D+01  0.31732538086D+01

  THE BC MATRIX
-0.16294768664D+01
 0.18976447418D+00
-0.23923620636D+01
 0.36761772931D+01
 0.13407481212D+01
 0.15725120366D+01
 0.22340357463D+00
-0.69029606181D+00

  THE CC MATRIX  (COLUMNS    1 -    4)
 0.35961167081D+01  0.15175212634D+01  0.85487098833D+01 -0.10248679150D+02

  THE CC MATRIX  (COLUMNS    5 -    8)
-0.97262926290D+01  0.20239432618D+01  0.72133650879D+01  0.52914248276D+00
```

Bibliography

[AS65] Athans, M. and F.C. Schweppe. "Gradient Matrices and Matrix Calculations", Tech. Note 1965-63, M.I.T. Lincoln Laboratory, Lexington, MA, November 1965.

[BH89] Bernstein, D.S. and W.H.Haddad. "LQG Control with an H_∞ Performance Bound: A Riccati Equation Approach", *IEEE Trans. Auto. Control*, Vol. 34, No. 3: 293-305 (March 1989).

[Dai90] Dailey, R.L. "Lecture Notes for the Workshop on H_∞ and μ Methods for Robust Control", in conjunction with *American Control Conference*, San Diego, CA, (21-22 May 1990).

[DGKF89] Doyle, J.C., K.Glover, P.P.Khargonekar and B.A.Francis. "State-Space Solutions to Standard H_2 and H_∞ Control Problems", *IEEE Trans. Auto. Control*, Vol. 34, No. 8: 831-847 (August 1989).

[Doy78] Doyle, J.C. "Guaranteed Margins for LQG Regulators", *IEEE Trans. Auto. Control*, Vol. 23, No. 4: 756-757 (August 1978).

[Fox71] Fox, R.L. *Optimization Methods for Engineering Design*, Reading, MA: Addison-Wesley, 1971.

[KJ72] Kreindler, E. and A. Jameson. "Conditions for Nonnegativeness of Partitioned Matrices", *IEEE Trans. Auto. Control*, Vol. 17, No. 2: 147-148 (February 1972).

[KR91] Khargonekar, P.P. and M.A.Rotea. "Mixed H_2/H_∞ Control: A Convex Optimization Approach", *IEEE Trans. Auto. Control*, Vol. 36, No. 7: 824-837 (July 1991).

[MG90] Mustafa, D. and K. Glover. *Minimum Entropy H_∞ Control.* in Lecture Notes in Control & Information Sciences. Berlin: Springer-Verlag, 1990.

[RB86] Ridgely, D.B. and S.S.Banda. *Introduction to Robust Multivariable Control*, AFWAL-TR-85-3102. Flight Dynamics Lab, Wright-Patterson Air Force Base, OH, February 1986.

[Rid91a] Ridgely, D.B. *A Nonconservative Solution to the General Mixed H_2/H_∞ Optimization Problem*. PhD dissertation. Massachusetts Institute of Technology, MA, 1991.

[Rid91b] Ridgely, D.B. Class notes in MECH 623, Advanced Function Optimization. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, Spring 1991.

[RK91] Rotea, M.A. and P.P.Khargonekar. "H_2-Optimal Control with an H_∞-Constraint: The State Feedback Case", *Automatica*, Vol. 27, No. 2: 307-316 (March 1991).

[SZ72] Snyders, J. and M. Zakai. "On Nonnegative Solutions of the Equation $AD + DA^T = -C^*$". *SIAM Journal of Appl. Math.*, Vol. 18, No. 3: 704-417 (May 1970)

[Won85] Wonham, W.M. Linear Multivariable Control: A Geometric Approach. New York: Springer-Verlag, 1985.

[YBC90] Yeh, H.H., S.S.Banda and B.C.Chang, "Necessary and Sufficient Conditions for Mixed H_2 and H_∞ Optimal Control", *Proc. of 29th Conference on Decision and Control*. 1013-1017. Honolulu, HI, December 1990.

[Zam66] Zames, G. "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems, Part I: Conditions Derived Using Concepts of Loop Gain, Conicity, and Positivity", *IEEE Trans. Auto. Control*, Vol. 11, No. 4: 228-238 (April 1966).

[ZDGB90] Zhou, K., J.Doyle, K.Glover and B.Bodenheimer, "Mixed H_2 and H_∞ Control", *Proc. of American Control Conference*. 2502-2507. San Diego, CA, May 1990.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Master's Thesis

4. TITLE AND SUBTITLE
INVESTIGATION OF THE EFFECTS OF INCREASED ORDER COMPENSATORS IN MIXED H_2/H_∞ OPTIMIZATION

5. FUNDING NUMBERS

6. AUTHOR(S)
Scott R. Wells, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GAE/ENY/91D-14

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Siva Banda
WL/FIGC
Wright-Patterson AFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The problem of minimizing the 2-norm of one transfer function subject to an ∞-norm bound on another transfer function is examined for increased order controllers. In particular, the theoretical results of the full order case are extended to the higher order case, and SISO and MIMO numerical examples are given for increasingly higher order compensators. Some of the key proofs for higher order compensators include: the global minimum 2-norm is unachievable under output feedback for certain levels of γ regardless of compensator order; the solution to the mixed H_2/H_∞ problem lies on the boundary of the ∞-norm constraint for this same range of γ's; and the suboptimal mixed problem converges to the optimal in the limit for higher order controllers. Also, it is shown that the optimal compensator order for the mixed H_2/H_∞ problem is greater than the order of the plant under certain conditions, and a conjecture about the optimal order for the mixed problem is made.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Optimization, Optimal Control, Mixed H_2/H_∞, Increased Order Controllers, Control Theory, Transfer Functions	185
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL